

# Touch Sensing Library

## Obsługa bezstykowych przycisków pojemnościowych

*Interfejs użytkownika występuje w większości urządzeń elektronicznych. Jest to element bardzo ważny, a jednak jego istotność jest często przez konstruktorów niedoceniana. Dobrze wykonany interfejs użytkownika pozwala na komfortowe, intuicyjne i zarazem niezawodne użytkowanie urządzenia. Jednym z najistotniejszych komponentów składających się na interfejs użytkownika są przyciski, które są podstawowym elementem umożliwiającym interakcję użytkownika z urządzeniem.*

Przyciski przez długi okres opierały się postępowi techniki i reprezentowane były niezmiennie przez podzespoły mechaniczne, których działanie opiera się na mechanizmie stykowym. Obecny rozwój technologii spowodował znaczny postęp w tej dziedzinie, dzięki czemu obok klasycznych rozwiązań pojawiły się nowe, dotąd nieznanne. Najpowszechniejszą z nich jest metoda bezstykowa, oparta na pomiarze pojemności. Ma ona wiele zalet, wśród których wymienić można np. niskie koszty, łatwość implementacji, dużą trwałość. Producenci układów scalonych, widząc te zalety, wychodzą naprzeciw oczekiwaniom, oferując zintegrowane sterowniki klawiatur pojemnościowych. Wśród największych producentów zajmujących się tą branżą znajdują się między innymi firmy: Freescale, Microchip NXP, Atmel, Elan i STMicroelectronics. Produkty tych firm opisane zostały w EP 11/2009, EP 1/2010 oraz EP 5/2010. Firma STMicroelectronics, kontynuując prace nad przyciskami pojemnościowymi, przedstawiła bibliotekę Touch Sensing Library przeznaczoną dla mikrokontrolerów z rodziny STM8. Narzędzie to pozwala na implementację w mikrokontrolerze nawet zaawansowanego interfejsu złożonego z przycisków pojemnościowych przy użyciu standardowych zasobów mikrokontrolera i jednocześnie bez konieczności użycia zewnętrznego kontrolera.

### Czujniki pojemnościowe i zasada ich działania

Zasada działania bezstykowych klawiatur pojemnościowych opiera się na zjawisku zmiany ilości gromadzonego ładunku elektrycznego w elektrodzie (będącej polem przyciskowym) w zależności od odległości dzielącej elektrodę i ludzkie ciało. Zbliżenie palca powoduje wprowadzenie dodatkowej pojemności. W dużym przybliżeniu pojem-

ność spodziewaną w warunkach oddziaływania ludzkiego palca na zaprojektowane pole przycisku obliczyć można za pomocą wzoru opisującego pojemność kondensatora płaskiego:

$$C = \frac{\epsilon_R \epsilon_0 A}{d}$$

gdzie:

$C$  – pojemność

$\epsilon_R$  – względna przenikalność elektryczna ośrodka, z którego wykonano dzielący okładki izolator

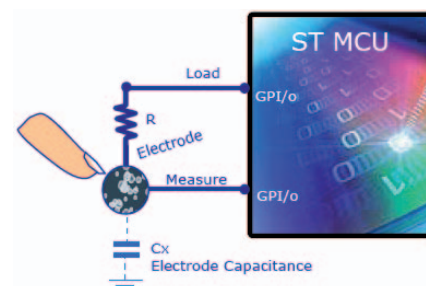
$\epsilon_0$  – przenikalność elektryczna wolnej przestrzeni

$A$  – pole powierzchni okładek

$d$  – odległość okładek kondensatora

W kontekście przycisku pojemnościowego  $\epsilon_R$  jest przenikalnością elektryczną materiału separującego pole przycisku,  $\epsilon_0$  jest przenikalnością elektryczną powietrza,  $A$  jest powierzchnią pola przycisku, natomiast  $d$  odległością między polem przycisku a palcem. Za pomocą powyższego wzoru zinterpretować można ponadto sposób zmian pojemności – wraz ze zmniejszaniem dystansu dzielącego palec i pole przycisku pojemność będzie rosła, natomiast oddalanie palca spowoduje zmniejszenie pojemności [2].

W celu maksymalnego uproszczenia sposobu odczytu stanów przycisków firma STMicroelectronics zaproponowała do tego celu metodę, której realizacja możliwa jest za pomocą podstawowych zasobów spotykanych w mikrokontrolerach. Wymaganymi peryferiami są linie ogólnego przeznaczenia w liczbie odpowiadającej liczbie przycisków oraz dwa układy licznikowe. Sposób pomiaru polega na okresowym ładowaniu i rozładowywaniu pojemności elektrody. Proces ładowania i rozładowania jest odmierzany przez układ licznikowy i na podstawie zmie-

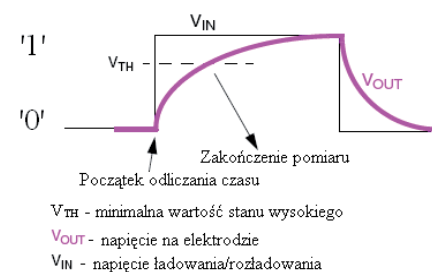


zonego czasu określany jest aktualny stan przycisku. W momencie startu timera na wszystkie elektrody podawany jest przez rezystor stan wysoki. W miarę upływu czasu ilość gromadzonego ładunku zwiększa się i napięcie na elektrodach wzrasta. W momencie odczytania na danej linii sygnałowej stanu wysokiego obliczany jest czas, który minął od rozpoczęcia ładowania (rys. 1).

Po sprawdzeniu w ten sposób wszystkich linii od pól przycisków pojemność elektrod jest rozładowywana przez podanie na linie stanu niskiego. Następnie cały proces sprawdzenia przycisków wykonuje się cyklicznie poprzez powtarzanie opisanych etapów. Schemat elektryczny przedstawiający sposób podłączania przycisków zamieszczony został na rys. 3.

### Obsługa biblioteki Touch Sensing Library

Biblioteka Touch Sensing Library jest kompleksowym narzędziem do obsługi



**Rys. 1.** Proces ładowania i rozładowania pojemności elektrody oraz zmieniający się poziom napięcia na elektrodzie [3] Jeśli zmierzony czas przekroczył ustaloną wartość, oznacza to, że cykl ładowania był wydłużony z powodu dodatkowej pojemności wprowadzonej przez ludzkie ciało, a więc przycisk został naciśnięty (rys. 2)

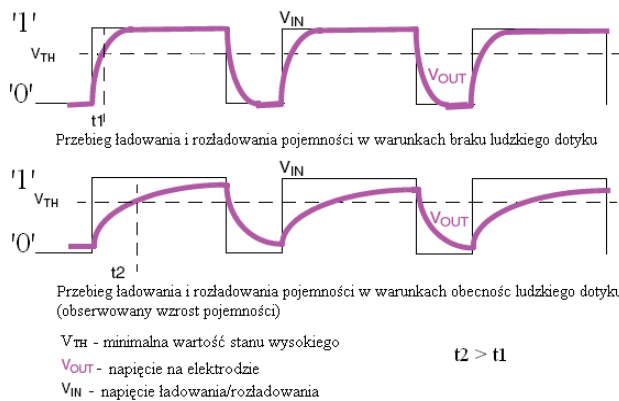
Tab. 1. Przykładowe przyporządkowanie wybranych peryferii do elementów wykonawczych biblioteki Touch Sensing Library [1]

| Funkcja                                       | Nazwa przy dyrektywie #define w pliku STM8_TSL_RC_Configuration.h  | Wartość  | Opis   |
|---|--|--|--|
| Wybór mikrokontrolera                         | STM8S  | 1  | Wybrana podrodzina STM8S   |
| Licznik nr 1                                  | TIMACQ<br>TIMACQ_CNTR_ADD  | TIM3<br>0x5328   | Timer do licznika nr 1 = TIMER3<br>Adres timera = 0x5328   |
| Licznik nr 2                                  | TIMTICK  | TIM4   | Timer do licznika nr 2 = TIMER4  |
| Linia sygnałowa do ładowania/<br>rozładowania | LOADREF_PORT_ADDR<br>LOADREF_BIT   | GPIOC_BaseAddress<br>0x04  | Linia ładowania/rozładowania = PC2   |
| Przyciski jednocanałowe                       | SCKEY_P1_KEY_COUNT<br>SCKEY_P1_PORT_ADDR<br>SCKEY_P1_A<br>SCKEY_P1_DRIVEN_SHIELD_MASK<br>SCKEY_P2_COUNT<br>SCKEY_P3_COUNT  | 1<br>GPIOC_BaseAddress<br>0x02<br>0x08<br>0<br>0                             | Liczba klawiszy = 1<br>Port przycisku = port C<br>Pin przycisku = pin nr 1<br>Warstwa ochronna dla przycisku P1<br>wyłączona<br>Przycisk P2 wyłączony<br>Przycisk P3 wyłączony |
| Przyciski wielokanałowe                       | NUMBER_OF_MULTI_CHANNEL_KEYS   | 0  | Funkcja wielokanałowych przycisków<br>wyłączona  |
| Maski elektrod                                | GPIOA_ELECTRODES_MASK<br>GPIOA_ELECTRODES_MASK<br>GPIOA_ELECTRODES_MASK<br>GPIOA_ELECTRODES_MASK<br>GPIOA_ELECTRODES_MASK<br>GPIOA_ELECTRODES_MASK<br>GPIOA_ELECTRODES_MASK<br>GPIOA_ELECTRODES_MASK<br>GPIOA_ELECTRODES_MASK<br>GPIOA_ELECTRODES_MASK | 0x00<br>0x00<br>0x0A<br>0x00<br>0x00<br>0x00<br>0x00<br>0x00<br>0x00<br>0x00 | Definicja masek dla elektrod (wartość<br>nieużywanych masek = 0x00)  |

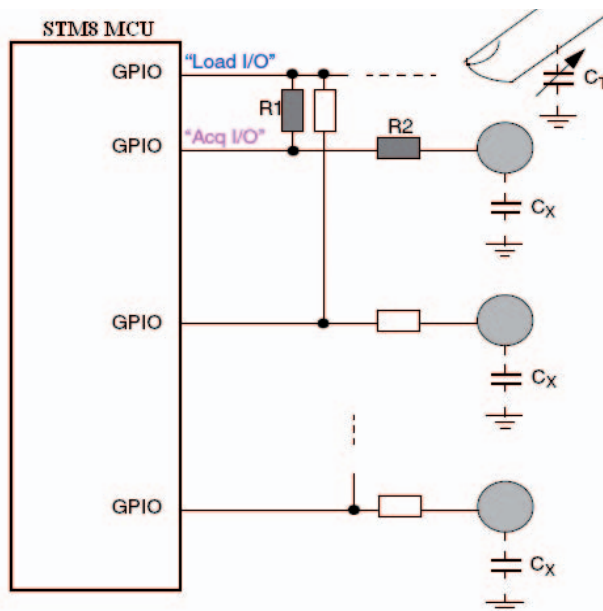
bezstykowych przycisków pojemnościowych. Umożliwia nie tylko odczyt stanów przycisków metodą opisaną powyżej, ale oferuje ponadto szereg mechanizmów zwiększających funkcjonalność tego procesu. Są to między innymi zaimplementowane algorytmy pełniące funkcje filtrów przeciwzakłóceń oraz wbudowane procedury kalibrujące. Pozwala to na wykrycie błędów związanych z tak zwanymi fałszywymi naciśnięciami, a więc błędami odczytu wynikającymi ze zmiany pojemności elektrody wywołanej np. zakłóceniami pochodzącymi od zasilania czy zmianami środowiskowymi (temperatura, wilgotność, kurz).

Sama architektura biblioteki ma budowę warstwową. Można w niej wyróżnić zasadniczo trzy bloki:

- Acquisition layer – warstwa odpowiedzialna za ładowanie/rozładowanie elektrod i zbieranie surowych danych
- Post processing layer – warstwa kontrolująca



Rys. 2. Proces sprawdzania stanu przycisków (różnica przebiegów dla samej elektrody oraz elektrody ze zbliżonym palcem) [3]



Rys. 3. Sposób podłączania przycisków pojemnościowych do układu mikrokontrolera [3]

ca kalibrację, filtrowanie i adaptację do zmian środowiskowych

- API – warstwa umożliwiająca użytkownikowi konfigurację i dostęp do danych wynikowych oraz zmiennych statusowych

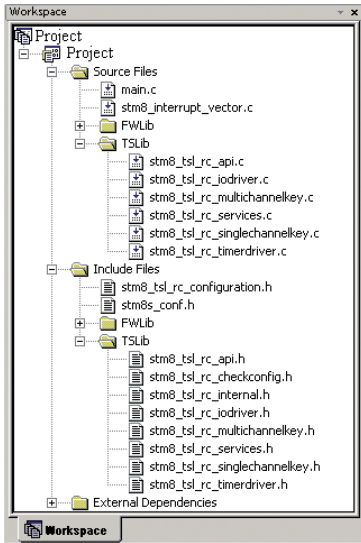
Struktura plików biblioteki podzielona została ze względu na realizowane funkcje. Podział ten został przeprowadzony zgodnie z opisanymi warstwami, a ponadto został rozszerzony o elementy charakterystyczne dla mikrokontrolera, a więc np. plik służący do wyboru stosowanej podrodziny STM8, plik konfiguracyjny wykorzystywanych peryferii czy plik definiujący liczbę użytych przycisków oraz ich przyporządkowanie do portów. Kompletna struktura plików przedstawiona została na rys. 4.

Pracę z biblioteką Touch Sensing Library należy rozpocząć od edycji pliku STM8\_TSL\_RC\_Configuration.h. Należy w nim zdefiniować peryferie, które będą używane przez bibliotekę oraz wskazać porty podłączone do pól przycisków i linii ładowania/rozładowania. Przykładowa konfiguracja przedstawiona jest w tab. 1. Wykorzystanymi układami licznikowymi są Timer3 oraz Timer4. Port PC2 odpowiedzialny jest za ładowanie i rozładowanie pojemności, natomiast do portu PC1 doprowadzone jest pole przycisku pojemnościowego (rys. 5).

Inicjalizacji biblioteki dokonuje się za pomocą wywołania funkcji TSL\_Init(). Aktywacja przycisku wykonywana jest poprzez wypełnienie struktury konfiguracyjnej sSCKeyInfo:

```
sSCKeyInfo[0].
Setting.b.IMPLEMENTED = 1;
sSCKeyInfo[0].Setting.b.ENABLED
= 1;
sSCKeyInfo[0].DESGroup = 0x01;
```

Wykrycie zmiany stanu przycisków możliwe jest poprzez monitorowanie wartości



Rys. 4. Widok plików składających się na bibliotekę Touch Sensing Library w przykładowym projekcie w środowisku ST Visual Develop

pół struktury TSL\_GlobalSetting. Przykładowy kod wykonujący tą czynność może wyglądać następująco:

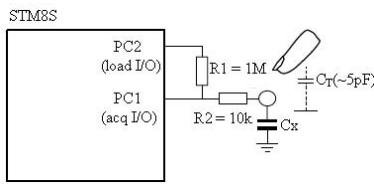
```
if (TSL_GlobalSetting.b.CHANGED)
{
    TSL_GlobalSetting.b.CHANGED = 0;
    if (sSCKeyInfo[0].
        Setting.b.DETECTED)
    {
        /* wykrycie wciśnięcia przycisku */
    }
    TSL_Action();
}
```

Kompleksowy program obsługujący przyciski pojemnościowe powinien być zgodny ze schematem blokowym z rys. 6.

Kod programu realizujący czynności zgodne ze schematem blokowym:

```
#include "stm8s.h"
#include "stm8_tsl_rc_api.h"

void main(void)
{
    /* Konfiguracja zegara */
    CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV1);
    /* Inicjalizacja biblioteki "Touch Sensing Library" */
    TSL_Init();
    /* Aktywacja wybranego przycisku */
    sSCKeyInfo[0].Setting.b.IMPLEMENTED = 1;
}
```



Rys. 5. Połączenie mikrokontrolera STM8S z układem RC przycisku pojemnościowego

```
sSCKeyInfo[0].Setting.b.ENABLED = 1;
sSCKeyInfo[0].DESGroup = 0x01;
/* Włączenie przerw */
enableInterrupts();

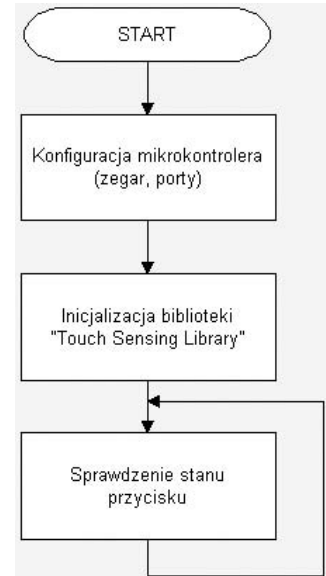
while(1)
{
    if (TSL_GlobalSetting.b.CHANGED )
    {
        TSL_GlobalSetting.b.CHANGED = 0;
        if (sSCKeyInfo[0].Setting.b.DETECTED)
        {
            /* wykryto wciśnięcie przycisku */
            TSL_Action();
        }
    }
}
```

Dokładny opis wszystkich aspektów dotyczących biblioteki Touch Sensing Library znajduje się w pliku pomocy STM8 TSL Manual. Zostały tam opisane między innymi wszystkie funkcje i struktury danych wchodzące w skład tego narzędzia. Pakiet instalacyjny biblioteki zawierający ten dokument oraz pliki źródłowe biblioteki wraz z przykładami dostępny jest pod poniższym adresem:

[http://www.st.com/stonline/products/support/micro/files/stm8s\\_touch\\_sensing\\_package.zip](http://www.st.com/stonline/products/support/micro/files/stm8s_touch_sensing_package.zip)

**Podsumowanie**

Biblioteka Touch Sensing Library jest kolejnym krokiem w rozwoju dziedziny elektroniki, jaką są bezstykowe przyciski pojemnościowe. Narzędzie do ich obsługi zaproponowane przez firmę STMicroelectronics jest ciekawym rozwiązaniem, gdyż umożliwia programową implementację obsługi przycisków w mikrokontrolerach rodziny STM8. Jakość i funkcjonalność tej metody obsługi przyci-



Rys. 6. Schemat blokowy przykładowego programu do obsługi przycisków pojemnościowych

sków nie ustępuje rozwiązaniom sprzętowym, a więc zintegrowanym kontrolerom klawiatur pojemnościowych. Dodatkowym atutem jest możliwość nieodpłatnego korzystania z tego narzędzia, gdyż jest ono udostępniane przez producenta za darmo. Daje to możliwość szybkiego i taniego zaadaptowania tego rozwiązania w nowych lub istniejących konstrukcjach.

**Szymon Panecki**  
szymon.panecki@pwr.wroc.pl

*Literatura*

- [1] [www.st.com](http://www.st.com) Adjustable LED blinking speed using STM8S-DISCOVERY touch sensing key, UM0833 User manual
- [2] [www.st.com](http://www.st.com) Guidelines for designing touch sensing applications, AN2869 application note
- [3] [www.st.com](http://www.st.com) RC acquisition principle for touch sensing applications, AN2927 application note
- [4] [www.st.com](http://www.st.com) STM8S Access Line/Performance Line, datasheet

R E K L A M M A