

# STMPE811

## Kontroler rezystancyjnego panelu dotykowego

Coraz więcej otaczających nas urządzeń elektronicznych sterowanych jest za pośrednictwem dotyku. Technologia ta w niektórych dziedzinach zaczyna wręcz wypierać typowe rozwiązania w postaci klawiatury czy myszy. Na co dzień można ją spotkać choćby w telefonach komórkowych, odtwarzaczach MP3, komputerach przenośnych czy automatach biletowych, a jest to zaledwie garstka przykładów. Wyposażenie nowokonstruowanych urządzeń w panele dotykowe jest jednocześnie coraz łatwiejsze, ponieważ na rynku nietrudno o gotowe układy sterujące. Jednym z przykładów takich sterowników jest układ STMPE811 oferowany przez ST Microelectronics.



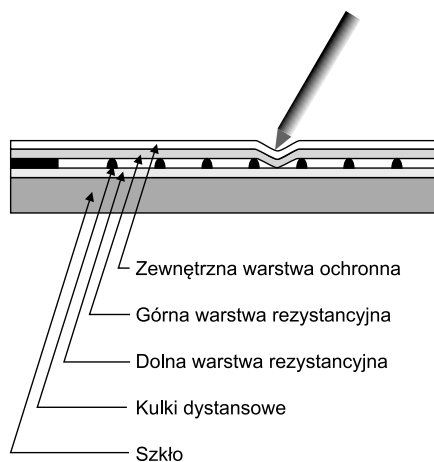
**Dodatkowe materiały na CD/FTP:**

<ftp://ep.com.pl>, user: 13057, pass: 41sjv430

Wśród paneli dotykowych dominują dwie technologie – pojemnościowa i rezystancyjna. Panele pojemnościowe reagują na zmianę pojemności, wywołaną np. dotknięciem powierzchni panelu palcem. Ich główną zaletą jest duża czułość (reagują nawet na lekkie dotknięcia) oraz pochłanianie niewielkiej ilości światła pochodzącego z leżącego najczęściej pod nimi wyświetlacza. Wśród wad wymienić należy wyższą cenę, ograniczone możliwości obsługi np. rysikiem czy ręką w rękawiczce oraz podatność na zabrudzenia i wilgoć.

### Panel rezystancyjny – zasada działania

Panele rezystancyjne zbudowane są z kilku warstw (**rysunek 1**), z których najważniejsze są dwie cienkie warstwy podatnej na ugięcia folii, pokrytej rezystywną warstwą przewodzącą. Pomiędzy warstwami znajduje się niewielka szczelina, a w niej nieprzewodzące mikrokulki dystansowe. W momencie dotknięcia panelu, warstwy stykają się i tworzą dzielnik napięciowy.



Rysunek 1. Budowa panelu rezystancyjnego

Napięcia mierzone w obu warstwach są proporcjonalne do pozycji dotkniętego punktu, co pozwala wyznaczyć jego położenie. Aby zapewnić całości konstrukcji odpowiednią sztywność i wytrzymałość, folia jest umieszczana na grubszym podłożu ze szkła lub polietylenu. Podstawowe zalety paneli rezystancyjnych to możliwość obsługi nie tylko palcem, ale także dowolnym rysikiem, odporność na zabrudzenia, możliwość pomiaru zarówno pozycji jak i siły nacisku oraz działanie w szerokim zakresie temperatur. Znaczącą zaletą jest także niższy (w porównaniu z panelami pojemnościowymi) koszt produkcji. Wśród wad wymienić należy mniejszą przezroczystość w porównaniu do paneli pojemnościowych, konieczność wywierania nacisku, by wywołać reakcję panelu oraz dłuższy czas reakcji. To wszystko sprawia, że panele rezystancyjne lepiej nadają się m.in. do urządzeń pracujących w trudniejszych warunkach np. poza budynkami lub w warunkach przemysłowych. Mimo tego można je spotkać w wielu innych zastosowaniach, np. w telefonach komórkowych.

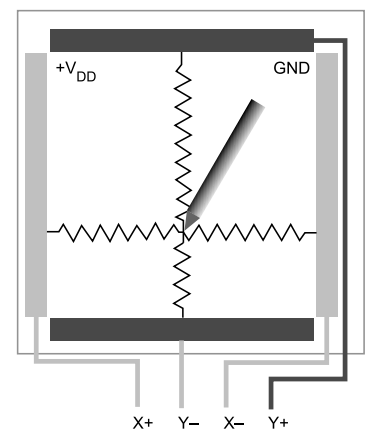
Określanie współrzędnych dotknięcia w panelach rezystancyjnych jest wykonywane na kilka sposobów, z użyciem od 4 do 8 przewodów. W tym opisie skupimy się tylko na tej drugiej metodzie, co wynika z możliwości układu STMPE811. W celu określenia współrzędnej X sterownik panelu przykładą napięcie  $V_{DD}$  do linii X+ oraz napięcie 0 V do linii X-. Jednocześnie przetwornik AC mierzy napięcie na linii Y+ (**rysunek 2**). Następnie, podobna procedura jest wykorzystywana do ustalenia współrzędnej Y. Biorą w niej udział linie Y+, Y-, a napięcie jest mierzone na linii X. W rezultacie otrzymujemy odczyty napięcia, których wartości są proporcjonalne do miejsca dotknięcia i na podstawie, których określane są obie jego

współrzędne. Aby zapewnić dokładny pomiar pozycji dotknięcia oraz wykrywanie ruchu, cały proces jest nieustannie powtarzany, ze znaczną częstotliwością.

Jak widać, określanie współrzędnych punktu dotknięcia nie jest samo w sobie trudne i można by je zrealizować programowo z wykorzystaniem przetworników AC wbudowanych w mikrokontroler. Jednak uzyskanie dużej dokładności wymaga częstego wykonania opisanej procedury, co mogłoby pochłonąć znaczną część jego mocy obliczeniowej. Dlatego warto scedować to zadanie na wyspecjalizowany układ, taki jak np. STMPE811.

### Podstawowe możliwości STMPE811

Układ STMPE811 jest kontrolerem rezystancyjnego panelu dotykowego i wykorzystuje do określania pozycji dotknięcia metodę 4-przewodową. Jego najważniejszą część stanowi 12-bitowy przetwornik analogowo-cyfrowy oraz układ przełączania linii XY. Przetwornik może pracować również w trybie 10-bitowym. Do komunikacji z układem można użyć interfejsu SPI lub I<sup>2</sup>C. Wybór następuje poprzez podanie niskiego lub wysokiego stanu napięcia na linii IN1 (doprowadzenie 9 obudowy). Na osobnej linii (INT, nóżka 2) STMPE811 może zgłaszać do mikrokontrolera sygnały żądania przerwania. Ponadto, układ może służyć również jako ekspander portu GPIO o 8 liniach wejścia-wyjścia. Linie te pokrywają się jednak m.in. z liniami X i Y, nie można więc wówczas jednocześnie wykorzystywać układu do sterowania panelem



Rysunek 2. Odczyt współrzędnej X

dotykowym. Dodatkowo, układ wyposażony jest w czujnik temperatury. Pamięć układu podzielona jest na kilkadziesiąt rejestrów, z których najważniejsze zamieszczono w tabeli 1.

## Inicjalizacja układu

W przypadku wykorzystania do komunikacji z układem interfejsu I<sup>2</sup>C może on być widoczny pod adresem 0x82 lub 0x88. Wybór zależy od stanu linii A0 (doprowadzenie

3 obudowy). Zgodnie z zaleceniami producenta, kolejność operacji przy konfiguracji układu do pracy jako kontroler panelu dotykowego jest następująca:

- Włączenie sygnałów taktujących do wybranych bloków układu, tj. czujnika temperatury, portów GPIO, kontrolera panelu dotykowego i przetwornika AC. Do samej obsługi panelu wystarczy włączyć taktowanie tylko dwóch ostatnich bloków.
- Włączenie wybranych sygnałów przerwań.
- Wybranie czasu konwersji, trybu pracy i źródła napięcia odniesienia dla przetwornika AC.
- Wybranie częstotliwości taktowania przetwornika AC.
- Wyłączenie alternatywnych funkcji portów GPIO.
- Wybór parametrów wykrywania dotyku: liczby uśrednień pomiarów, czasu opóźnienia przed wykrywaniem dotknięcia i czasu ustalania sygnału sterującego liniami X i Y.
- Ustalenie progowej wartości zapewnienia rejestrów FIFO, po przekroczeniu, której układ będzie zgłaszał przerwanie.
- Wyzerowanie zawartości rejestrów FIFO.
- Wybór formatu danych dla osi Z.
- Wybór prądu sterującego liniami X i Y.
- Wybór trybu pomiaru i włączenie wykrywania dotyku.
- Wyczyszczenie zgłoszeń przerwań.
- Włączenie przerwań.

Na **listingu 1** zamieszczono funkcję *STMPE\_I2C\_Init()* napisaną w języku C, realizującą przedstawioną procedurę.

Warto w tym miejscu dodać jeszcze parę słów o wyborze parametrów wykrywania dotyku. Liczba uśrednień określa, ile kolejnych pomiarów ma zostać wykonanych i następnie uśrednionych zanim sterownik poda współrzędne punktu dotknięcia. Dostępne są wartości 1, 2, 4 i 8. Ta funkcja sterownika pozwala zmniejszyć wpływ szumów pomiarowych na dokładność odczytu pozycji. Czas trwania całego jednego cyklu pomiarowego zależy od czasu opóźnienia, jaki upływa od momentu rozpoczęcia cyklu do podania napięcia na linię sterującą, czasu ustalania sygnału sterującego oraz czasu próbkowania. Czas trwania cyklu można wyznaczyć następująco, np.: dla pomiaru współrzędnych X, Y, Z:

$$\begin{aligned} & \text{Opóźnienie} \times 2 + \text{Czas ustalania} \times 3 \\ & + (\text{Czas próbkowania} \times \text{Liczba uśrednień}) \times 3, \\ & \text{natomiast dla pomiaru współrzędnych X i Y:} \\ & \text{Opóźnienie} \times 2 + \text{Czas ustalania} \times 2 \\ & + (\text{Czas próbkowania} \times \text{Liczba uśrednień}) \times 2. \end{aligned}$$

Tworząc własny projekt poszczególne parametry czasowe należy dobrać do zastosowanego panelu. Ich wartości zależą w szczególności od wymiarów panelu i charakterystyk elektrycznych linii X i Y (np. dołączonych kondensatorów filtrujących).

Tabela 1. Najważniejsze rejestry układu STMPE811

| Nazwa rejestru | Adres | Liczba bitów | Znaczenie  |
|----------------|-------|--------------|--|
| CHIP_ID        | 0x00  | 16           | Identyfikator układu (wartość 0x0811)  |
| SYS_CTRL1      | 0x03  | 8            | Reset softwareowy; tryb hibernacji   |
| SYS_CTRL2      | 0x04  | 8            | Aktywacja sygnałów taktujących poszczególne bloki układu STMPE811                    |
| INT_CTRL       | 0x09  | 8            | Konfiguracja sygnałów przerwań   |
| INT_EN         | 0x0A  | 8            | Włączenie sygnałów przerwań  |
| INT_STA        | 0x0B  | 8            | Stan i identyfikacja źródeł poszczególnych przerwań                                  |
| TSC_CTRL       | 0x40  | 8            | Konfiguracja sterownika panelu dotykowego  |
| TSC_CFG        | 0x41  | 8            | Konfiguracja sterownika panelu dotykowego  |
| WDW_TR_X       | 0x42  | 16           | Współrzędna X prawego, górnego narożnika aktywnego okna                              |
| WDW_TR_Y       | 0x44  | 16           | Współrzędna Y prawego, górnego narożnika aktywnego okna                              |
| WDW_BL_X       | 0x46  | 16           | Współrzędna X lewego, dolnego narożnika aktywnego okna                               |
| WDW_BL_Y       | 0x48  | 16           | Współrzędna Y lewego, dolnego narożnika aktywnego okna                               |
| FIFO_TH        | 0x4A  | 8            | Poziom zapewnienia FIFO wywołujący przerwanie  |
| FIFO_STA       | 0x4B  | 8            | Aktualny stan FIFO   |
| FIFO_SIZE      | 0x4C  | 8            | Aktualne zapewnienie FIFO  |
| TSC_DATA_X     | 0x4D  | 16           | Współrzędna X miejsca dotknięcia   |
| TSC_DATA_Y     | 0x4F  | 16           | Współrzędna Y miejsca dotknięcia   |
| TSC_DATA_Z     | 0x51  | 8            | Współrzędna Z miejsca dotknięcia (siła nacisku)                                      |
| TSC_FRACTION_Z | 0x56  | 8            | Konfiguracja formatu danych dla współrzędnej Z                                       |
| TSC_DATA       | 0x57  | 8            | Połączone współrzędne miejsca dotknięcia   |
|                | 0xD7  | 8            | Połączone współrzędne miejsca dotknięcia – odczyt w trybie autoinkrementacji adresów |
| TSC_I_DRIVE    | 0x58  | 8            | Wybór prądu sterującego liniami X i Y  |
| TEMP_CTRL      | 0x60  | 8            | Konfiguracja czujnika temperatury  |
| TEMP_DATA      | 0x61  | 16           | Wynik pomiaru temperatury  |

Listing 1. Funkcja *STMPE\_I2C\_Init()*

```
char STMPE811_I2C_Init(void)
{
    volatile long int i;
    char Buffer[2]={0};
    int RegVal;
    //zerowanie układu
    STMPE811_WriteRegister(STMPE811_SYS_CTRL1, 0x02);
    //sygnały taktujące: cz. temperatury - ON, GPIO - OFF,
    //kontroler panelu dotykowego - ON, ADC - ON
    STMPE811_WriteRegister(STMPE811_SYS_CTRL2, 0x04);
    //włączenie przerwan: przepelnienie FIFO, dotyk
    STMPE811_WriteRegister(STMPE811_INT_EN, 0x03);
    //czas konwersji ADC = 80 taktów, tryb 12-bit,
    //wewnętrzne źródło napięcia odniesienia
    STMPE811_WriteRegister(STMPE811_ADC_CTRL1, 0x49);
    for (i=0;i<500000ul;i++);
    //taktowanie ADC = 3.25MHz
    STMPE811_WriteRegister(STMPE811_ADC_CTRL2, 0x01);
    //alternatywne funkcje GPIO - wylaczone
    STMPE811_WriteRegister(STMPE811_GPIO_AF, 0x00);
    //usrednianie pomiarow - 4 próbki; opoznienie wykrycia dotyku - 1ms,
    //czas ustalania prądu sterującego - 1ms
    STMPE811_WriteRegister(STMPE811_TSC_CFG, 0xA3);
    //prog FIFO = 1
    STMPE811_WriteRegister(STMPE811_FIFO_TH, 0x01);
    //reset FIFO
    STMPE811_WriteRegister(STMPE811_FIFO_STA, 0x01);
    STMPE811_WriteRegister(STMPE811_FIFO_STA, 0x00);
    //Format danych osi Z = z,zzzzzz
    STMPE811_WriteRegister(STMPE811_TSC_FRACTION_Z, 0x07);
    //prad sterujacy liniami panelu - 50mA
    STMPE811_WriteRegister(STMPE811_TSC_I_DRIVE, 0x01);
    //wlacz panel, pomiar XY
    STMPE811_WriteRegister(STMPE811_TSC_CTRL, 0x03);
    //wyczysc zgloszenia przerwan
    STMPE811_WriteRegister(STMPE811_INT_STA, 0xFF);
    //wlacz przerwania
    STMPE811_WriteRegister(STMPE811_INT_CTRL, 0x01);
    //kontrolny odczyt ID układu - powinno byc 0x0811
    STMPE811_ReadRegisters(STMPE811_CHIP_ID, 2, Buffer);
    RegVal=Buffer[0]<<8|Buffer[1];
    if (RegVal == 0x0811) return STMPE811_OK;
    return STMPE811_ERROR;
}
```

Tabela 2. Formaty danych w rejestrze TSC\_DATA

| Tryb pomiaru | Liczba bajtów do odczytu z TSC_DATA | Bajt 0   | Bajt 1              | Bajt 2  | Bajt 3  |
|--------------|-------------------------------------|----------|---------------------|---------|---------|
| XYZ<br>000   | 4                                   | X [11:4] | X [3:0]<br>Y [11:8] | Y [7:0] | Z [7:0] |
| XY<br>001    | 3                                   | X [11:4] | X [3:0]<br>Y [11:8] | Y [7:0] |         |
| X<br>010     | 2                                   | X [11:4] | X [3:0]             |         |         |
| Y<br>011     | 2                                   | Y [11:4] | Y [3:0]             |         |         |
| Z<br>100     | 1                                   | Z [7:0]  |                     |         |         |

## Odczyt danych

Kontroler po uruchomieniu i skonfigurowaniu działa autonomicznie. Bada stan panelu i gdy wykryje, że został on dotknięty, umieszcza odczytaną pozycję w buforze FIFO. Maksymalnie, bufor ten pomieścić może 128 odczytów składających się ze składowej X, Y i Z. Zapisane wartości można odczytać na kilka sposobów, z kilku różnych rejestrów. Np. rejestry TSC\_DATA\_X, TSC\_DATA\_Y i TSC\_DATA\_Z pozwalają odczytać każdą składową osobno, natomiast rejestr TSC\_DATA zwraca wyniki w postaci spakowanej, zależnej od wybranego trybu pomiaru. Taka forma przedstawiania danych pozwala na przyspieszenie ich odczytu. Np. jeśli interesuje nas tylko pomiar współrzędnych X i Y to zamiast odczytu dwóch rejestrów po 2 bajty wystarczy odczytać 3 bajty z jednego rejestru. Zyskuje się przy tym także na liczbie operacji związanych z obsługą samego transferu, gdyż nie trzeba osobno odwoływać się do poszczególnych rejestrów. Możliwe tryby pomiaru i formaty zapisu danych w rejestrze TSC\_DATA przedstawia tabela 2. Na listingu 2 przedstawiono z kolei kod funkcji *STMPE911\_ReadXYZ()* odczytującej dane, przekształcającej je z postaci spakowanej do zwykłej i umieszczającej je

w tablicy współrzędnych. Wykorzystana wewnątrz funkcja *STMPE811\_ReadNRegisters()* realizuje natomiast odczyt zadanej liczby kolejnych bajtów poczynając od zadanego adresu. Odczytane wartości umieszczane są w pomocniczej tablicy – buforze.

Podawane przez układ wartości mieszczą się w zakresie od 0 do 4095 lub od 0 do 1024 co wynika z ustawionej podczas konfiguracji rozdzielczości przetwornika AC.

## Określanie aktywnego obszaru panelu i automatyczne wykrywanie ruchu

Oprócz określania pozycji dotknięcia panelu, kontroler ma dwie inne przydatne funkcje. Pierwsza z nich to możliwość definiowania aktywnego okna. Dzięki temu można ustalić, że układ będzie reagował na dotknięcia tylko w wybranym obszarze panelu, a dotknięcia poza nim będą ignorowane. Można to wykorzystać np. gdy program oczekuje od użytkownika dotknięcia panelu w konkretnym miejscu. Pozycję okna można dowolnie zmieniać podczas pracy układu. Na listingu 3 przedstawiono funkcję *STMPE811\_SetActiveWindow()*, która ustawia aktywne okno. Parametrami jej wywołania są współrzędne X i Y lewego dolnego i prawego górnego narożnika okna.

Drugą z funkcji jest automatyczne wykrywanie ruchu. Dzięki niej można tak skonfigurować sterownik, by nowe pozycje podawane były tylko wówczas, gdy różnią się od pozycji poprzedniej. W tym celu wyznaczony jest indeks opisany zależnością:

$$\text{indeks ruchu} = (\text{aktualne X} - \text{poprzednio podane X}) + (\text{aktualne Y} - \text{poprzednio podane Y}).$$

Jeżeli wartość indeksu przekroczy zadany poziom, kontroler zwraca nową pozycję. Dostępnych jest 8 różnych poziomów progowych indeksu: 0, 4, 8, 16, 32, 64, 92 i 127. Wykorzystanie tej funkcji pozwala znacznie zmniejszyć ilość danych przekazywanych przez kontroler, ponieważ nie są one przesyłane wtedy, gdy pozycja punktu dotknięcia nie zmienia się lub zmienia w niewielkim zakresie. Warto dodać, że w wypadku uwzględniania pomiaru siły nacisku (współrzędnej Z) zwiększenie nacisku powoduje zwrócenie pozycji niezależnie od ustawień wykrywania ruchu.

## Sygnalizacja rozpoznania dotyku i przerwania

Jak wspomniano wcześniej, układ, poprzez linię INT, może zgłaszać do mikrokontrolera żądania przerwania. Są one aktywowane w rejestrze INT\_EN, którego kolejne bity odpowiadają za następujące przerwania:

- bit 7 – przerwania od GPIO
- bit 6 – przerwania od ADC
- bit 5 – przekroczenie zadanego progu temperatury
- bit 4 – FIFO puste
- bit 3 – FIFO zapelnione
- bit 2 – przepełnienie FIFO
- bit 1 – przekroczenie progu zapelnienia FIFO
- bit 0 – zmiana stanu dotyku.

Takie samo znaczenie i kolejność mają poszczególne bity rejestru INT\_STA, w któ-

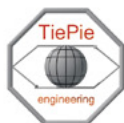
REKLAMA

## TiePieSCOPE HS805 – przystawka oscyloskopowa 1GS/s z generatorem

Moduł był testowany i został opisany w *Elektronice Praktycznej 12/2011*



- DSO: 2 wejścia BNC
- próbkowanie do 1GS/s (1 kanał), 500MS/s (2 kanały)
- pasmo 250MHz (-3dB)
- rozdzielczość 8 bitów
- zakresy napięć 200mV...80V
- sprzęganie wejścia AC, DC
- impedancja wejściowa 1MΩ / 20pF
- zabezpieczenie wejść ±200V
- pamięć 32MS/kanal
- AWG: 1 wyjście BNC
- maksymalne próbkowanie 200MS/s
- pasmo 20MHz
- rozdzielczość 14 bitów dla 200MS/s
- pamięć 32MS
- przebiegi: sinus, trójkąt, prostokąt, DC, szumy, zdefiniowany
- funkcje: oscyloskop, generator, analizator widma, woltomierz, rejestrator, analizator protokołów
- interfejs USB 2.0 High Speed / 1.1 Full Speed



Egmont Instruments, ul. Chłodna 39, pawilon 11, 00-867 Warszawa  
tel. 228506205, 692501750, faks 226540248  
e-mail tiepie@egmont.com.pl, http://www.egmont.com.pl/tiepie

**Listing 2. Funkcja *STMPE911\_ReadXYZ()***

```
void STMPE811_ReadXYZ(char mode, int * XYZval){
//Odczyt wartosci XYZ w zalezności od trybu (mode [0-4])
//XYZval należy zadeklarować jako tablice 3-elementowa typu int
char Buffer[4]={0};
switch (mode){
case 0:{ //wartosci XYZ
STMPE811_ReadNRegisters(STMPE811_TSC_DATA_AI, 4, Buffer);
XYZval[0]=(int) (Buffer[0]<<4) |(Buffer[1]>>4);
XYZval[1]=(int) (Buffer[1]&0x0F)<<8 |Buffer[2];
XYZval[2]=Buffer[3];
} break;
case 1:{ //wartosci XY
STMPE811_ReadNRegisters(STMPE811_TSC_DATA_AI, 3, Buffer);
XYZval[0]=(int) (Buffer[0]<<4) |(Buffer[1]>>4);
XYZval[1]=(int) (Buffer[1]&0x0F)<<8 |Buffer[2];
XYZval[2]=0;
} break;
case 2:{ //tylko X
STMPE811_ReadNRegisters(STMPE811_TSC_DATA_AI, 2, Buffer);
XYZval[0]=(int) (Buffer[0]<<4) |(Buffer[1]>>4);
XYZval[1]=0;
XYZval[2]=0;
} break;
case 3:{ //tylko Y
STMPE811_ReadNRegisters(STMPE811_TSC_DATA_AI, 2, Buffer);
XYZval[0]=0;
XYZval[1]=(int) (Buffer[0]<<4) |(Buffer[1]>>4);
XYZval[2]=0;
} break;
case 4:{ //tylko Z
STMPE811_ReadNRegisters(STMPE811_TSC_DATA_AI, 1, Buffer);
XYZval[0]=0;
XYZval[1]=0;
XYZval[2]=Buffer[0];
} break;
}
}
```

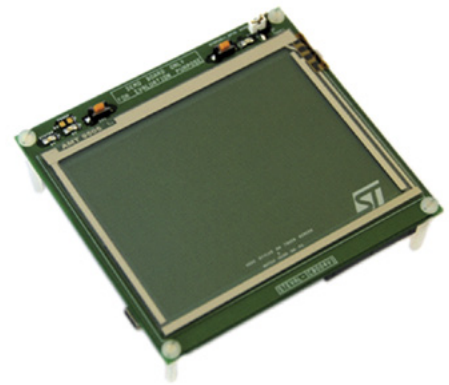
**Listing 3. Funkcja *STMPE811\_SetActiveWindow()***

```
void STMPE811_SetActiveWindow(int BL_X, int BL_Y, int TR_X, int TR_Y){
//Ustawianie współrzędnych aktywnego okna
//BL_X - lewy, dolny X, BL_Y - lewy, dolny Y
//TR_X - prawy, gorny X, TR_Y - prawy, gorny Y
//prawy gorny X
STMPE811_WriteRegister(STMPE811_WDW_TR_X_LO, (char) (TR_X>>8) &0x000F);
STMPE811_WriteRegister(STMPE811_WDW_TR_X_HI, (char) (TR_X) &0x00FF);
//prawy gorny Y
STMPE811_WriteRegister(STMPE811_WDW_TR_Y_LO, (char) (TR_Y>>8) &0x000F);
STMPE811_WriteRegister(STMPE811_WDW_TR_Y_HI, (char) (TR_Y) &0x00FF);
//lewy dolny X
STMPE811_WriteRegister(STMPE811_WDW_BL_X_LO, (char) (BL_X>>8) &0x000F);
STMPE811_WriteRegister(STMPE811_WDW_BL_X_HI, (char) (BL_X) &0x00FF);
//lewy dolny Y
STMPE811_WriteRegister(STMPE811_WDW_BL_Y_LO, (char) (BL_Y>>8) &0x000F);
STMPE811_WriteRegister(STMPE811_WDW_BL_Y_HI, (char) (BL_Y) &0x00FF);
}
```

rym przechowywane są informacje o źródłach aktualnie zgłoszonego żądania. Jak więc widać, zgłoszenie może być wywołane przez kilka różnych zdarzeń. Z punktu widzenia obsługi panelu dotykowego najważniejsze są dwa ostatnie, tj. przekroczenie progu FIFO oraz zmiana stanu dotyku.

Rozpoznanie dotknięcia sygnalizowane jest w ten sposób, że po jego wykryciu, ustawiany jest bit 7 rejestru TSC\_CTRL. Bit ten jest z kolei zerowany wtedy, gdy dotyk nie został wykryty. Jednocześnie każda zmiana stanu tego bitu sygnalizowana jest w rejestrze stanu przerwań INT\_STA na bicie 0. Oznacza to, że przerwanie zgłaszane jest zarówno w momencie dotknięcia jak i zwolnienia dotyku panelu. Z kolei na bicie 1 tego rejestru sygnalizowane jest przekroczenie zadane- go w rejestrze FIFO\_TH progu zapalenia kolejki FIFO. Jeżeli próg ten jest równy 1, wówczas sygnał przerwania będzie zgłaszany po odczytaniu każdej nowej pozycji. Jeśli zaś próg będzie wyższy, przerwanie będzie zgłaszane rzadziej, a w procedurze jego obsługi będzie należało odczytać kilka kolejnych pozycji zgromadzonych w kolejce FIFO.

Rejestr INT\_STA, oprócz odczytu źródła sygnału przerwania pozwala także na kaso-



**Rysunek 3. Moduł STEVAL-ICB004V1**

wanie zgłoszeń. W tym celu, na pozycji bitu odpowiadającego za dane źródło należy zapisać wartość 1. Zapisanie 0 nie wywołuje żadnych skutków.

Warto zauważyć, że stan poszczególnych bitów rejestru INT\_STA zmieniany jest niezależnie od tego, czy dane źródło zostało aktywowane w rejestrze INT\_EN, czy nie. Dzięki temu można np. sprawdzić stan kolejki FIFO nawet, jeśli układ nie ma zgłaszać przerwań. W przypadku wykorzystania mechanizmu przerwań w obsłudze układu STMPE811 konieczne jest jeszcze napisanie funkcji obsługi tego przerwania po stronie mikrokontrolera. Na **listingu 4** przedstawiono przykład takiej funkcji dla mikrokontrolera STM32. Sygnał INT ze sterownika podłączony został do linii 5 portu GPIOB jako przerwanie zewnętrzne.

Na początku procedury należy sprawdzić źródło przerwania. Jeśli jest to przepełnienie kolejki FIFO, następuje odczyt jednego kompletu współrzędnych punktu dotknięcia i umieszczenie ich w tablicy XYZ. Jeśli jest to zmiana stanu dotknięcia – zmieniana jest wartość globalnej zmiennej *Dotyk* oraz stan diody sygnalizującej stan panelu. W kolejnym kroku zerowany jest w kontrolerze przerwań mikrokontrolera bit stanu obsługi przerwania zewnętrznego, a następnie, zależnie od odczytanego wcześniej źródła przerwania, zerowane są w układzie STMPE811 odpowiednie bity rejestru INT\_STA. Dalszą obsługą odczytanych informacji (np. wyświetleniem odczytanej pozycji) zajmuje się główna część programu.

**Środowisko testowe**

Do testowania sterownika wykorzystano moduł STEVAL-ICB004V1 (**rysunek 3**), który składa się z panelu dotykowego, układu STMPE811 oraz mikrokontrolera STM32F103. Rolą mikrokontrolera jest pośrednictwo pomiędzy sterownikiem a komputerem PC, gdzie moduł podłączyć można do portu USB. Do modułu dołączony jest na płycie CD program demonstrujący jego działanie. Ponieważ oglądanie gotowych rozwiązań jest mniej pouczające niż samodzielne ich napisanie, autor zdecydował

| ZL27ARM – JP5 (GPIOB) | STEVAL ICB004V1 – J4 |
|-----------------------|----------------------|
| PB6                   | Pin 1 – STMPE811_SCL |
| PB7                   | Pin 2 – STMPE811_SDA |
| PB5                   | Pin 9 – TSCREEN_INT  |
| GND                   | Pin 3 – TS_GND       |
| +3,3V                 | Pin 4 – TS_VDD       |

Uwaga  
linie SCL, SDA i INT muszą być połączone z V<sub>DD</sub> (+3,3V) poprzez rezystory podciągające 4,7kΩ.

**Listing 4. Funkcja obsługi przerwania**

```

typedef enum {FALSE = 0, TRUE = !FALSE} bool;
extern volatile int XYZ[3];
extern volatile bool Dotyk;
void EXTI9_5_IRQHandler(void)
{
    char cRegVal;
    if (EXTI_GetITStatus(EXTI_Line5) != RESET){
        STMPE811_ReadRegister(STMPE811_INT_STA, &cRegVal);
        //sa dane w FIFO - odczytaj pozycje
        if (cRegVal&0x02)
            STMPE811_ReadXYZ(1, XYZ);
        //opuszczenie lub podniesienie pisaka
        if (cRegVal&0x01) {
            Dotyk=!Dotyk;
            GPIO_WriteBit(GPIOB, GPIO_Pin_8,
                (BitAction)(1-GPIO_ReadOutputDataBit(GPIOB, GPIO_Pin_8)));
        }
        EXTI_ClearITPendingBit(EXTI_Line5);
        //kasowanie przerw w układzie
        if ((cRegVal&0x03)==3) {
            STMPE811_WriteRegister(STMPE811_INT_STA, 0x03);
        } else {
            if (cRegVal&0x01)
                STMPE811_WriteRegister(STMPE811_INT_STA, 0x01);
            if (cRegVal&0x02)
                STMPE811_WriteRegister(STMPE811_INT_STA, 0x02);
        }
    }
}
    
```

się na podłączenie sterownika do innego mikrokontrolera. Wybór padł na zestaw ZL27ARM również z mikrokontrolerem STM32F103. Zmiana taka była ułatwiona, ponieważ moduł STEVAL-ICB004V1 pozwala odłączyć część związaną z układem STMPE811 od mikrokontrolera znajdującą się razem z nim na płycie. Rozłączenie obu układów polega na ustawieniu na spodniej stronie modułu STEVAL wszyst-

kich zworek z grupy *Microcontroller to touch controller interface* w pozycję NC. Następnie, do odpowiednich wyprowadzeń złącza J4 podłączyć można linie łączące sterownik z zewnętrznym mikrokontrolerem. Wykaz połączeń pomiędzy modułem a zestawem ZL27ARM podany jest w **tabeli 3**.

Obserwacje działania panelu wykazały, że kontroler nie zwraca pełnego zakre-

su wartości. Współrzędne zawierały się w przedziale od około 150 do około 3950, co oznacza, że nawet przy krańcowych położeniach punktu dotknięcia, nie są zwracane skrajne wartości współrzędnych. Nie powinno to jednak stanowić większego problemu, ponieważ wystarczy odpowiednio, programowo skalibrować odczyty. Należy mieć również na uwadze, że panel dotykowy najczęściej umieszczony jest przed jakimś ekranem. Oznacza to, że odczytywane współrzędne dotknięcia trzeba i tak zwykle dodatkowo przeliczyć na współrzędne ekranu.

W materiałach dołączonych do niniejszego artykułu znajdują się dwa programy demonstracyjne. Pierwszy z nich zawiera obsługę sterownika STMPE811 z wykorzystaniem przerw, zaś drugi – uproszczoną, bezprzerwanową. Oba programy realizują odczyt pozycji dotknięcia i jej podanie na wyświetlaczu alfanumerycznym, sygnalizowanie dotknięcia, obsługę okna (aktywacja przyciskiem SW0 w ZL27ARM) oraz rozpoznawanie ruchu (aktywacja przyciskiem SW1 w ZL27ARM). Dodatkowo, na początku pokazywany jest identyfikator układu oraz aktualna zmierzona przez niego temperatura.

**Marek Galewski**  
 marg@mech.pg.gda.pl

REKLAMA

**Altium Designer**

„Przetestowaliśmy narzędzia wszystkich wiodących dostawców oprogramowania EDA, w poszukiwaniu idealnego rozwiązania, które pozwoli dostarczać projekty naszym klientom tak szybko, jak to tylko możliwe. Dzięki uniwersalności, elastyczności i łatwości użycia, system Altium był bezkonkurencyjny.”

Phil Gibson  
 Wiceprezes National Semiconductor

**evatronix**

ul. Przybyły 2, 43-300 Bielsko-Biała, tel. 33 499 59 00, 499 59 12  
 eda@evatronix.com.pl, www.evatronix.com.pl/eda