

Obsługa rezystancyjnych paneli dotykowych z AR1020

Kilka miesięcy temu na łamach EP prezentowałem artykuł na temat obsługi rezystancyjnych paneli dotykowych za pomocą przetwornika A/C. Niestety, takie rozwiązanie zabiera część mocy obliczeniowej CPU, gdyż musi ono cyklicznie sprawdzać rezystancję panelu. Dlatego warto do tego zadania zastosować specjalizowany układ scalony. Jednym z takich układów jest kontroler dotykowych paneli rezystancyjnych AR1020 produkcji firmy Microchip.

AR1020 umożliwia dołączenie 4-ro, 5-cio i 8-mio przewodowych, rezystancyjnych paneli dotykowych, o rezystancji warstwy 50...2000 Ω . Rozdzielczość pomiaru dla każdej z osi wynosi 10 bitów, co jest wystarczające dla większości zastosowań. Z systemem nadrzędnym układ komunikuje się za pomocą interfejsów I²C lub SPI, a żądanie obsługi transmisji może być zgłaszane przez wyjście przerwania IQR. Układ jest zasilany napięciem 3,3...5 V. Charakteryzuje go bardzo mały pobór energii (w automatycznym trybie Standby prąd pobierany z zasilania jest mniejszy od 1 μ A!).

Zastosowanie układu kontrolera zwalnia programistę od realizacji wielu trudnych zadań. Należy do nich np. kalibracja panelu dotykowego. Układ może generować do 140 raportów pomiarowych na sekundę. Sposób odpowiedzi sterownika na zdarzenie dotknięcia panelu dotykowego jest w pełni konfigurowalny. Dodatkowo, układ jest wyposażony w 128 bajtów pamięci EEPROM dostępnej z poziomu aplikacji sterującej. Można jej użyć np. do przechowywania nieulotnych nastaw.

Na **rysunku 1** przedstawiono rozmieszczenie wyprowadzeń układu AR1020 w obudowie SOIC lub SSOP.

Wybór jednego z dwóch interfejsów I²C lub SPI jest dokonywany sprzętowo za pomocą poziomu napięcia na doprowadzeniu M1. W tym przykładzie użyto I²C. Układ AR1020 pracuje w trybie Slave z 7-bitowym adresem równym dla operacji zapisu 9AH lub 9BH dla operacji odczytu. Komunikacja jest dwukierunkowa, a więc mikrokontroler nadrzędny może wysyłać rozkazy sterujące oraz raporty pomiarowe.

W trybie zapisu komunikacja z układem AR1020 odbywa się następująco:

I²C Start \rightarrow Adres=9AH \rightarrow 00H \rightarrow 0x55 \rightarrow Liczba bajtów danych \rightarrow Komenda \rightarrow Dana(1)...Dana(n) \rightarrow I²C Stop

Odbiór każdego bajtu jest potwierdzany sygnałem I²C ACK. Liczba bajtów danych jest powiększona o bajt komendy. Dane są opcjonalne i przesyłane tylko wtedy, gdy wymaga tego komenda.

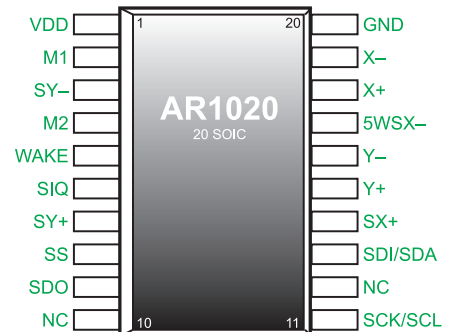
Po wysłaniu rozkazu sterującego przez układ host (mikrokontroler) układ AR1020 każdorazowo potwierdza odebranie przesłanej ramki transmisji. Potwierdzenie to zawiera status wykonania ostatnio przesłanego rozkazu i jest poprzedzane przez ustawienie wyprowadzenia IRQ (domyślnie ma ono poziom niski). Po tym host powinien odczytać przesyłane dane. Ich format jest następujący: I²C Start \rightarrow 9BH \rightarrow 0x55 \rightarrow Liczba bajtów danych \rightarrow Status \rightarrow Komenda \rightarrow Dana(1)...Dana(n) \rightarrow I²C Stop

Liczba bajtów danych jest powiększona o bajt statusu i bajt komendy. Pole „komenda” zawiera zwrócony kod rozkazu sterującego, którego można użyć do identyfikacji potwierdzanego polecenia. Bajt statusu może przyjąć następujące wartości:

- 0: polecenie zostało wykonane,
- 1: nieznanne polecenie,
- 3: nieznanny nagłówek ramki danych,
- 4: przekroczenie czasu na wysłanie kompletnej komendy (timeout=100 ms),
- FCH: anulowano procedurę kalibracji sterownika.

Przesłanie przez host sygnału NACK wymusza zakończenie transmisji. W podobny sposób AR1020 zgłasza naciśnięcie panelu dotykowego, przy czym ramka transmisji (raport pomiarowy) ma w tym przypadku skład się z:

I²C Start \rightarrow 9B \rightarrow PEN \rightarrow XL \rightarrow XH \rightarrow YL \rightarrow YH \rightarrow I²C Stop

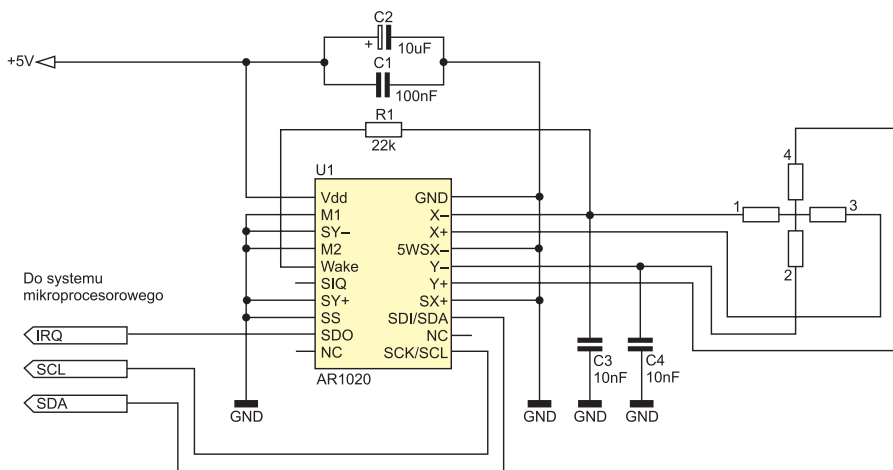


Pin	Symbol	Opis
1	Vdd	Napięcie zasilania
2	M1	Wybór interfejsu komunikacyjnego: GND \rightarrow I ² C, V _{dd} \rightarrow SPI
3	SY	Zacisk SY- dla 8-przewodowych paneli dotykowych. Połączyć z GND, gdy nie jest używany
4	M2	Wybór rodzaju panelu dotykowego: GND \rightarrow 4/8 przewodowy, Vdd \rightarrow 5 przewodowy
5	WAKE	Wybudzenie sterownika z trybu Standby
6	SIQ	Wyjście zgłoszenia przerwania dla interfejsu SPI lub wyjście sygnalizacyjne (diody LED) stanu pracy panela dotykowego. Pozostawić niepodłączone, jeśli nie jest używane
7	SY+	Zacisk SY+ dla 8-przewodowych paneli dotykowych. Połączyć z GND, gdy nie jest używany
8	SS	Sygnal SS (Slave Select) dla interfejsu SPI. Połączyć z GND, gdy nie jest używany
9	SDO	Wyjście danych interfejsu SPI lub wyjście IRQ (zgłoszenia przerwania) dla interfejsu I ² C
10, 12	NC	
11	SCK/SCL	Wejście sygnału zegarowego dla interfejsów SPI i I ² C
13	SDI/SDA	Wyjście/wejście danych dla interfejsu I ² C lub wejście danych dla interfejsu SPI
14	SX+	Zacisk SX+ dla 8-przewodowych paneli dotykowych. Połączyć z GND, gdy nie jest używany
15	Y+	Zacisk Y+ panelu dotykowego
16	Y-	Zacisk Y- panelu dotykowego
17	5WSX-	Zacisk 5W dla 5-przewodowych paneli dotykowych. Połączyć z GND, gdy nie jest używany
18	X+	Zacisk X+ panelu dotykowego
19	X-	Zacisk X- panelu dotykowego
20	GND	Masa zasilania

Rysunek 1. Rozmieszczenie wyprowadzeń AR1020

Tabela 1. Lista rozkazów układu AR1020

Kod rozkazu	Znaczenie	Opis
12H	Enable Touch	Włączenie obsługi panelu dotykowego.
13H	Disable Touch	Wyłączenie obsługi panelu dotykowego.
14H	Calibrate Mode	Inicjacja trybu kalibracji sterownika.
20H	Register Read	Odczyt rejestru sterującego.
21H	Register Write	Zapis do rejestru sterującego.
22H	Register Start Address Request	Żądanie przesłania adresu początku przestrzeni adresowej rejestrów sterujących.
23H	Registers Write to EEPROM	Zapis konfiguracji sterownika do pamięci EEPROM.
28H	EEPROM Read	Odczyt pamięci EEPROM w zakresie danych użytkownika.
29H	EEPROM Write	Zapis do pamięci EEPROM w zakresie danych użytkownika.
2BH	EEPROM Write to Register	Odczyt pamięci konfiguracji.



Rysunek 2. Typowa aplikacja układu AR1020 z 4-przewodowym panelem rezystancyjnym

PEN jest znacznikiem informującym o naciśnięciu lub zwolnieniu panelu dotykowego (w zależności od ustawień rejestru *Touch Mode Register*). Standardowo używa się bit0 znacznika PEN, który dla ustawień domyślnych jest ustawiony po naciśnięciu i wyzerowany po zwolnieniu panelu. Bajty XL i XH informują o umownej współrzędnej X miejsca naciśnięcia (mniej i bardziej znaczący bajt), natomiast YL i YH o współrzędnej Y. Rozdzielczość pomiaru wynosi 12 bitów, przy czym dla sterownika nieskalibrowanego dwa najmniej znaczące bity są równe 0, co determinuje 10-bitową, rzeczywistą rozdzielczość pomiarową.

Używając operacji zapisu i odczytu możemy sterować pracą układu AR1020 poprzez wysyłanie rozkazów sterujących oraz modyfikację rejestrów odpowiedzialnych za parametry algorytmu pomiarowego i zachowanie się układu w przypadku naciśnięcia

panelu dotykowego. W tabeli 1 zamieszczono listę dostępnych rozkazów.

Rejestr *Touch Mode Register* decyduje o treści raportów pomiarowych dla trzech rodzajów zdarzeń: naciśnięcie (PD2...PD0), przesunięcie po (PM1...PM0) i zwolnienie (PU2...PU0) panelu dotykowego. Jest to praktycznie jedyny rejestr, którego wartości domyślne (inicjowane po włączeniu zasilania) musimy zmienić. Pozostałe rejestry są zoptymalizowane dla większości rodzajów rezystancyjnych paneli dotykowych w związku z czym nie ma potrzeby ich modyfikowania.

Zapis do rejestrów sterujących układu AR1020 podlega jednak pewnym regułom, które dotyczą sposobu ustalenia rzeczywistego, bezwzględnego adresu żądanego rejestru sterującego, ponieważ podane w dokumentacji wartości wskazują jedynie na offset adresu konkretnego rejestru w odniesieniu

do początku przestrzeni adresowej rejestrów sterujących. Aby ustalić rzeczywisty adres żądanego rejestru należy w pierwszej kolejności pozyskać adres początku przestrzeni adresowej rejestrów sterujących. Do tego celu przewidziano specjalną komendę *Register Start Address Request* (22H). Aby mieć jednak pewność, że wysyłanie komendy sterującej nie zostanie przerwane przez wysłanie raportu pomiarowego zaleca się zachowanie następującej sekwencji:

- 1) *Disable Touch*,
- 2) odczekanie 50 ms,
- 3) *Register Start Address Request*,
- 4) *Enable Touch*.

Przy założeniu, że kontroler AR1020 powinien zgłosić raport o naciśnięciu panelu dotykowego z bitem PEN.0 ustawionym, następnie monitorować dalsze naciskanie panelu wysyłając raporty z ustawionym bitem PEN.0, a po zwolnieniu nacisku nie powinien wysłać żadnego raportu, do rejestru *Touch Mode Register* należy wpisać liczbę 50H. Należy to zrobić następująco:

- 1) wysłać rozkaz *Disable Touch* i odczytać odpowiedź układu,
- 2) odczekanie 50 ms,
- 3) wysłać rozkaz *Register Start Address Request* i odczytać odpowiedź,
- 4) Zapisać liczbę 50H do rejestru *Touch Mode Register* (offset 0x0C) i odczytać odpowiedź,
- 5) wysłać rozkaz *Enable Touch* i odczytać odpowiedź.

Przykład aplikacji

Na rysunku 2 przedstawiono schemat typowej aplikacji układu AR1020 z dołączonym 4-przewodowym panelem rezystancyjnym oraz interfejsem I²C używanym do komunikacji z systemem nadrzędnym.

Na listingu 1 zamieszczono procedurę konfigurującą wywoływaną na początku programu aplikacji użytkownika, która przesyła opisane wcześniej nastawy. Na listingu 2 zamieszczono procedurę obsługi przerwania zewnętrznego INT0 odpowiedzialną za odbiór raportów po naciśnięciu panelu dotykowego i zgłaszającą gotowość raportu do odbioru za pomocą bitu *Ready*. Obie procedury napisano w języku *Bascom*. Celowo nie są one zoptymalizowane pod kątem rozmiaru generowanego kodu wynikowego, ponieważ ich celem jest przejrzysta prezentacja sposobu obsługi programowej kontrolera AR120.

Listing 1. Prolog inicjalizacyjny programu obsługi sterownika AR1020

```

On Int0 Read_touch_report
Config Int0 = Rising
Set Portd.2
Declare Sub Initialize_ar1020
Declare Sub Write_ar1020(byval Command As Byte, Byval No_of_data As Byte)
Declare Sub Read_ar1020
Dim Global array(5) As Byte
Dim Btemp As Byte, Isrtemp As Byte
Dim Ready As Byte
Dim Xcord As Word, Ycord As Word
Irq Alias Pind.2

'Deklaracja procedury obsługi przerwania zewnętrznego INT0 - zgłoszenie raportu zdarzeń
'Konfiguracja przerwania zewnętrznego INT0 - przerwanie wyzwalane zbroczem rosnącym
'Podciągnięcie wejścia INT0 do plusa zasilania poprzez wewnętrzny rezystor podciągający
'Procedura inicjalizacyjna sterownika AR1020
'Procedura zapisująca dane do sterownika AR1020
'Procedura odczytująca dane ze sterownika AR1020
'Tablica pomocnicza
'Zmienne pomocnicze typu Byte
'Flaga naciśnięcia panelu dotykowego ustawiana w procedurze INT0
'Współrzędne miejsca naciśnięcia panela dotykowego
'Alias dla wyprowadzenia INT0 - zgłoszenie raportu „Touch Report” przez układ AR1020

```

Listing 2. Procedura zapisu danych do układu AR1020

```

Sub Write_ar1020(byval Command As Byte , Byval No_of_data As Byte)
Local Index As Byte
    'I2cStart - 0x9A Ack - 0x00 Ack - 0x55 Ack - 0x01 Ack - Command Ack - Dana(1)Ack ... Dana(n)Ack - I2c Stop
    'Wysyłamy sygnał Start
    Twcr = &B10100100
    Bitwait Twcr.twint , Set
    'Ustawiony TWINT, TWSTA, TWEN
    'Czekamy, aż interfejs TWI wykona Start i zgłosi ten fakt ustawiając flagę TWINT
    'Wysyłamy adres układu AR1020 w trybie zapisu

    Twdr = &H9A
    Twcr = &B10000100
    Bitwait Twcr.twint , Set
    'Ustawiony TWINT, skasowany TWSTA i ustawiony TWEN
    'Czekamy, aż interfejs TWI wyśle Adres i zgłosi ten fakt ustawiając flagę TWINT
    'Wysyłamy bajt zerowy - według specyfikacji układu AR1020

    Twdr = &H00
    Twcr = &B10000100
    Bitwait Twcr.twint , Set
    'Ustawiony TWINT, skasowany TWSTA i ustawiony TWEN
    'Czekamy, aż interfejs TWI wyśle wspomniany bajt i zgłosi ten fakt ustawiając flagę TWINT
    'Wysyłamy nagłówek ramki danych - &H55

    Twdr = &H55
    Twcr = &B10000100
    Bitwait Twcr.twint , Set
    'Ustawiony TWINT, skasowany TWSTA i ustawiony TWEN
    'Czekamy, aż interfejs TWI wyśle nagłówek i zgłosi ten fakt ustawiając flagę TWINT
    'Wysyłamy informację o rozmiarze ramki danych: No_of_data + 1

    Twdr = No_of_data + 1
    Twcr = &B10000100
    Bitwait Twcr.twint , Set
    'Ustawiony TWINT, skasowany TWSTA i ustawiony TWEN
    'Czekamy, aż interfejs TWI wyśle bajt i zgłosi ten fakt ustawiając flagę TWINT
    'Wysyłamy rozkaz sterujący

    Twdr = Command
    Twcr = &B10000100
    Bitwait Twcr.twint , Set
    'Ustawiony TWINT, skasowany TWSTA i ustawiony TWEN
    'Czekamy, aż interfejs TWI wyśle rozkaz i zgłosi ten fakt ustawiając flagę TWINT
    'Jeśli do komendy mają być dołączone dodatkowe dane to wysyłamy je korzystając z globalnej zmiennej
    tablicowej
    'zawierającej właśnie te dane (muszą być wcześniej wpisane do tejże tablicy)

    If No_of_data <> 0 Then
        FoF Index = 1 To No_of_data
        Twdr = Global arYay(index)
        Twcr = &B10000100
        Bitwait Twcr.twint , Set
        'Ustawiony TWINT, skasowany TWSTA i ustawiony TWEN
        'Czekamy, aż interfejs TWI wyśle kolejną daną i zgłosi ten fakt ustawiając flagę TWINT
        Next Index
    End If
    'Inicjujemy wysłanie sygnału Stop
    Twcr = &B10010100
    'Ustawiony TWINT, TWSTO i TWEN
    'I NIE czekamy na nic bo po wysłaniu sygnału Stop nie jest ustawiana flaga TWINT

End Sub

```

Listing 3. Procedura odczytu danych z układu AR1020

```

I2CStop
Sub Read_ar1020
Local Index As Byte
Local Rozmiar As Byte
    'I2cstart - 0x9B Ack - 0x55 Ack - Rozmiar ramki Ack - Odpowiedź Ack - Komenda Ack - Dana(1)Ack ... Dana(n)Ack -
    'Wysyłamy sygnał Start
    Twcr = &B10100100
    Bitwait Twcr.twint , Set
    'Ustawiony TWINT, TWSTA, TWEN
    'Czekamy, aż interfejs TWI wykona Start i zgłosi ten fakt ustawiając flagę TWINT
    'Wysyłamy adres układu AR1020 w trybie odczytu inicjując tym samym tryb Master Receiver dla naszego interfejsu TWI

    Twdr = &H9B
    Twcr = &B10000100
    Bitwait Twcr.twint , Set
    'Ustawiony TWINT, skasowany TWSTA i ustawiony TWEN
    'Czekamy, aż interfejs TWI wyśle Adres i zgłosi ten fakt ustawiając flagę TWINT
    'Inicjujemy odbiór pierwszego bajta z układu AR1020: będzie to bajt nagłówek &H55, którego odczyt pomijamy
    Twcr = &B11000100
    Bitwait Twcr.twint , Set
    'Ustawiony TWINT, ustawiony TWEN i ustawiony TWEA (wysyłanie potwierdzeń ACK)
    'Czekamy na otrzymanie bajta danych sygnalizowane ustawieniem flagi TWINT
    'Inicjujemy odbiór drugiego bajta z układu AR1020: będzie to bajt świadczący o rozmiarze przesyłanej ramki
    danych
    Twcr = &B11000100
    Bitwait Twcr.twint , Set
    'Ustawiony TWINT, ustawiony TWEN i ustawiony TWEA (wysyłanie potwierdzeń ACK)
    'Czekamy na otrzymanie bajta danych sygnalizowane ustawieniem flagi TWINT
    Rozmiar = Twdr
    Index = 1
    'Odczytujemy rozmiar ramki danych

    Do
        'Odczytujemy kolejne, przesyłane przez układ AR1020 bajty danych
        'Inicjujemy odbiór kolejnego bajta z układu AR1020 przy czym dla bajta ostatniego nie wysyłamy sygnału ACK
    sygnałując, iż
        'mikrokontroler nie oczekuje już kolejnych bajtów danych - wynika to ze specyfikacji ramki układu AR1020

        If Index <> Rozmiar Then Twcr = &B11000100 Else Twcr = &B10000100
        Bitwait Twcr.twint , Set
        'Czekamy na otrzymanie bajta danych sygnalizowane ustawieniem flagi TWINT
        Global array(index) = Twdr
        'Zapisujemy odebrany bajt do tablicy globalnej
        Incr Index
    Loop Until Index > Rozmiar

    Twcr = &B10010100
    'Inicjujemy wysłanie sygnału Stop
    'Ustawiony TWINT, TWSTO i TWEN
    'I NIE czekamy na nic bo po wysłaniu sygnału Stop nie jest ustawiana flaga TWINT

End Sub

```

Listing 4. Procedura inicjalizacyjna układu AR1020

```

Sub Initialize_ar1020
    'Uruchomienie i konfiguracja interfejsu TWI. Częstotliwość magistrali ustawiamy na 40kHz
    'Można ustawić większą częstotliwość transmisji jeśli magistrala posiada odpowiednią pojemność.

    Twbr = 92
    Twsr.1 = 0 , TWPS1
    Twsr.0 = 0 , TWPS0

    'Wysyłanie konfiguracji do układu AR1020
    'Wysyłanie rozkazu: „Disable Touch” i odczytanie odpowiedzi, której wartość pomijamy
    Call Write_ar1020(&H13 , &H00)
    Bitwait Irq , Set
    'Oczekiwanie na zgłoszenie odpowiedzi
    Call Read_ar1020
    Waitms 50

    'Wysyłanie rozkazu: „Register Start Address Request” i odczytanie odpowiedzi - ostatni, trzeci bajt odpowiedzi to
    'adres początku
    'przeźrzeni adresowej rejestrów sterujących
    Call Write_ar1020(&H22 , &H00)
    Bitwait Irq , Set
    'Oczekiwanie na zgłoszenie odpowiedzi
    Call Read_ar1020
    Btemp = Global_array(3)
    'Przechowanie adresu początku przestrzeni adresowej rejestrów sterujących
    'Zapis do rejestru: „Touch Mode Register” (offset 0x0C) wartości &H50 i odczytanie odpowiedzi, której wartość
    'pomijamy
    'Przygotowanie towarzyszących danych
    Global_array(1) = &H00
    'Starsza część bezwzględnej adresu rejestru „Touch Mode Register”
    Global_array(2) = Btemp + &H0C
    'Młodsza część bezwzględnej adresu rejestru „Touch Mode Register”
    Global_array(3) = &H01
    'Liczba rejestrów do zapisu
    Global_array(4) = &H50
    'Wartość zapisywanego rejestru
    Call Write_ar1020(&H21 , &H04)
    Bitwait Irq , Set
    'Oczekiwanie na zgłoszenie odpowiedzi
    Call Read_ar1020

    'Wysyłanie rozkazu: „Enable Touch” i odczytanie odpowiedzi, której wartość pomijamy
    Call Write_ar1020(&H12 , &H00)
    Bitwait Irq , Set
    'Oczekiwanie na zgłoszenie odpowiedzi
    Call Read_ar1020

    'Zezwolenie na obsługę przerwania zewnętrznego INT0 odpowiedzialnego za odbiór raportów o naciśnięciu panela
    'dotykowego
    Set Gifr.intf0
    'Wyczyszczenie ewentualnej flagi zgłoszenia przerwania INT0
    Enable Int0
    Enable Interrupts
End Sub

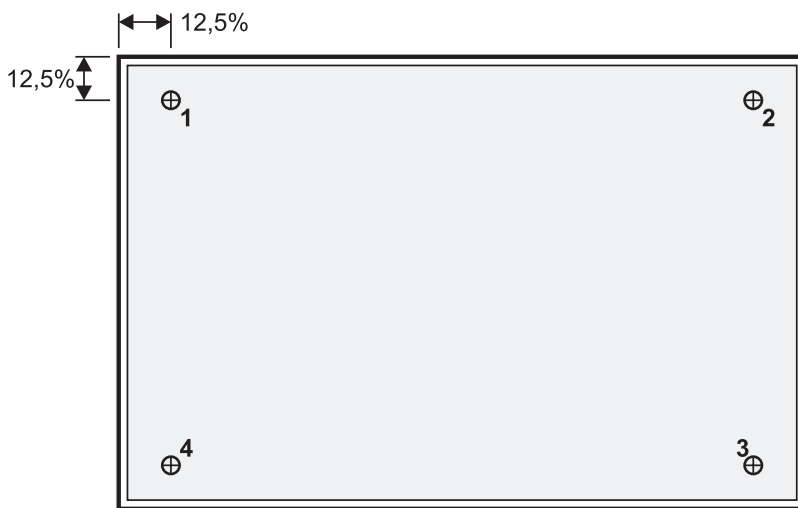
```

Listing 5. Procedura obsługi przerwania INTO odpowiedzialna za odbiór raportów zdarzeń

```

Read_touch_report:                'Zgłoszenie tego przerwania oznacza, iż układ AR1020 sygnalizuje chęć wysłania raportu pomiarowego w związku z
czyml                             'odczytujemy jego treść
                                   'Wysyłamy sygnał Start
    Twcr = &B10100100              'Ustawiony TWINT, TWSTA, TWEN
    Bitwait Twcr.twint , Set        'Czekamy, aż interfejs TWI wykona Start i zgłosi ten fakt ustawiając flagę TWINT
    TWI                             'Wysyłamy adres układu AR1020 w trybie odczytu inicjując tym samym tryb Master Receiver dla naszego interfejsu
    Twdr = &H9B
    Twcr = &B10000100              'Ustawiony TWINT, skasowany TWSTA i ustawiony TWEN
    Bitwait Twcr.twint , Set        'Czekamy, aż interfejs TWI wyśle Adres i zgłosi ten fakt ustawiając flagę TWINT
    Twcr = &B110000100              'Inicjujemy odbiór pierwszego bajta z układu AR1020: będzie to bajt PEN, którego odczyt pomijamy
    Bitwait Twcr.twint , Set        'Ustawiony TWINT, ustawiony TWEN i ustawiony TWEA (wysyłanie potwierdzeń ACK)
    Twcr = &B110000100              'Czekamy na otrzymanie bajta danych sygnalizowane ustawieniem flagi TWINT
    Bitwait Twcr.twint , Set        'Inicjujemy odbiór drugiego bajta z układu AR1020: będzie to bajt XL
    Twcr = &B110000100              'Ustawiony TWINT, ustawiony TWEN i ustawiony TWEA (wysyłanie potwierdzeń ACK)
    Bitwait Twcr.twint , Set        'Czekamy na otrzymanie przesyłanego bajta danych sygnalizowane ustawieniem flagi TWINT
    Twcr = &B110000100              'Odczytujemy bajt XL
    Bitwait Twcr.twint , Set        'Inicjujemy odbiór trzeciego bajta z układu AR1020: będzie to bajt XH
    Twcr = &B110000100              'Ustawiony TWINT, ustawiony TWEN i ustawiony TWEA (wysyłanie potwierdzeń ACK)
    Bitwait Twcr.twint , Set        'Czekamy na otrzymanie przesyłanego bajta danych sygnalizowane ustawieniem flagi TWINT
    Twcr = &B110000100              'Odczytujemy bajt XH
    Shift Xcord , Left , 8
    Xcord = Xcord Or Isrtemp        'W tym momencie w zmiennej Xcord znajduje się współrzędna X miejsca naciśnięcia panelu dotykowego
    Twcr = &B110000100              'Inicjujemy odbiór czwartego bajta z układu AR1020: będzie to bajt YL
    Bitwait Twcr.twint , Set        'Ustawiony TWINT, ustawiony TWEN i ustawiony TWEA (wysyłanie potwierdzeń ACK)
    Twcr = &B110000100              'Czekamy na otrzymanie przesyłanego bajta danych sygnalizowane ustawieniem flagi TWINT
    Bitwait Twcr.twint , Set        'Odczytujemy bajt YL
    Twcr = &B10000100              'Inicjujemy odbiór piątego i zarazem ostatniego bajta z układu AR1020: będzie to bajt YH
    Bitwait Twcr.twint , Set        'Ustawiony TWINT, ustawiony TWEN i wyzerowany TWEA (brak wysyłania potwierdzenia NACK)
    Twcr = &B10000100              'Czekamy na otrzymanie przesyłanego bajta danych sygnalizowane ustawieniem flagi TWINT
    Bitwait Twcr.twint , Set        'Odczytujemy bajt YH
    Twcr = &B10010100              'W tym momencie w zmiennej Ycord znajduje się współrzędna Y miejsca naciśnięcia panelu dotykowego
    Shift Ycord , Left , 8
    Ycord = Ycord Or Isrtemp        'Inicjujemy wysłanie sygnału Stop
    Twcr = &B10010100              'Ustawiony TWINT, TWSTO i TWEN
    Ready = 1                       'I NIE czekamy na nic bo po wysłaniu sygnału Stop nie jest ustawiana flaga TWINT
    Return                          'Ustawiamy flagę widoczną w pętli głównej aplikacji wskazująca na odebranie danych

```



Rysunek 3. Domyślne położenie punktów pomiarowych procedury kalibracyjnej

Kalibracja

Układ AR1020 jest wyposażony w specjalny mechanizm kalibracji niezbędny na przykład wtedy, gdy część aktywna wyświetlacza współpracującego z panelem dotykowym niecałkowicie pokrywa powierzchnię panelu lub gdy z czasem panel dotykowy ulegnie naturalnemu zużyciu. Procedura kalibracji umożliwia odpowiednie skalowanie obszaru roboczego. Podczas jej realizacji użytkownik musi podać czte-

ry wartości odniesienia, którymi są w tym przypadku określone, cztery miejsca naciśnięcia na panel dotykowy. Standardowo są to punkty znajdujące się w odległości 12,5% (odległość względna odniesiona do długości każdego z boków panelu) od każdego z rogów panelu. Jest to wartość domyślna, gdyż poprzez modyfikację rejestru *CalibratiInset Register* (offset 0x0E) można zmienić położenie punktów pomiarowych. Standardowe rozmieszczenie wspomnia-

nych punktów pomiarowych pokazano na **rysunku 3**. Procedura kalibracji jest bardzo dobrze znana, choćby z telefonów komórkowych wyposażonych w panel dotykowy. Tam wykonuje się ją dokładnie w taki sam sposób.

Przebieg procesu kalibracji przy założeniu domyślnej wartości rejestru *CalibratiInset Register* powinien być następujący:

- 1) wysłanie *Calibrate Mode*,
- 2) wyświetlenie na współpracującym wyświetlaczu LCD kolejnych 4 punktów kalibracyjnych i prośba o ich wskazanie przez użytkownika,
- 3) zakończenie procedury kalibracyjnej; sterownik AR1020 zapisze odpowiednie współczynniki w nieulotnej pamięci EEPROM jako wzorzec dla zastosowanego panelu dotykowego.

Podsumowanie

Należy podkreślić, że ten artykuł jest tylko dużym skrótem opisu możliwości, jakie drzemią w sterowniku AR1020. Mam jednak nadzieję, iż nawet te informacje zachęcą do samodzielnych prób z tym układem, ponieważ znacznie odciąża on mikrokontroler w aplikacji.

Robert Wołgajew, EP
robert.wolgajew@ep.com.pl

R E K L A M M A

forum.ep.com.pl