

STM8S001J3 (7)

Przetwornik A/C



Tematyką siódmej części cyklu artykułów dotyczących 8-pinowego mikrokontrolera STM8S001J3 jest przetwornik A/C (analogowo-cyfrowy). W artykule opisano budowę przetwornika oraz pokazano jak wykonać krok po kroku przykładową aplikację używającą tego zasobu mikrokontrolera.

Przetwornik A/C – wstęp

Często zdarza się, że system elektroniczny oparty na mikrokontrolerze wykorzystuje nie tylko sygnały cyfrowe, a więc takie, w których wyróżnić można wartość binarną 0 i 1 (np. sygnał PWM, interfejsy komunikacyjne I²C/UART/SPI itp.), ale używa również sygnałów analogowych, których wartość nie ogranicza się do logicznego 0 i 1, a zmienia się w sposób dowolny w zakresie między wartością minimalną i maksymalną. Do poprawnego odczytania wartości takich sygnałów mikrokontroler wykorzystuje dedykowany zasób, jakim jest przetwornik A/C. Jak sama nazwa wskazuje, można go zdefiniować jako blok konwertujący sygnał analogowy do postaci cyfrowej. Proces ten odbywa się w trzech etapach: próbkowania, kwantowania i kodowania. Próbkowanie to czynność zamiany sygnału ciąglego na sygnał dyskretny, a więc na grupę pomiarów wykonanych w stałym odstępie czasu. Kwantyzacja jest czynnością polegającą na przypisaniu każdej z próbek wartości liczbowej ze zbioru wartości, jakie przyjąć może wyjście przetwornika. Kodowanie natomiast jest zamianą wartości liczbowej wyjścia przetwornika na odpowiadającą jej sekwencję bitów. Poszczególne etapy tego procesu zilustrowano na **rysunku 1**.

Na przestrzeni czasu powstały różne architektury, według których działają przetworniki A/C. Wyróżnić można między innymi przetworniki o porównaniu bezpośrednim (*Flash*), przetworniki szeregowo-równoległe (*Two-Step Flash*), przetworniki potokowe (*Pipeline*), przetworniki całkujące (np. SAR – *Successive Approximation Register*) czy przetworniki Sigma-Delta. W mikrokontrolerach zazwyczaj spotyka się przetworniki A/C o architekturze SAR lub Sigma-Delta.

Jednym z najważniejszych parametrów przetworników jest ich rozdzielczość, a więc liczba bitów wyniku pomiaru. Porównując przetworniki popularnie integrowane w mikrokontrolerach należy podkreślić, że układy o architekturze SAR cechują się rozdzielczością na poziomie 8-18 bitów, co jest wartością typowo niższą od przetworników Sigma-Delta, które charakteryzują się rozdzielczością na poziomie 8...32 bitów.

Innym istotnym parametrem przetworników A/C jest częstotliwość próbkowania, a więc liczba pomiarów, które można wykonać w danej

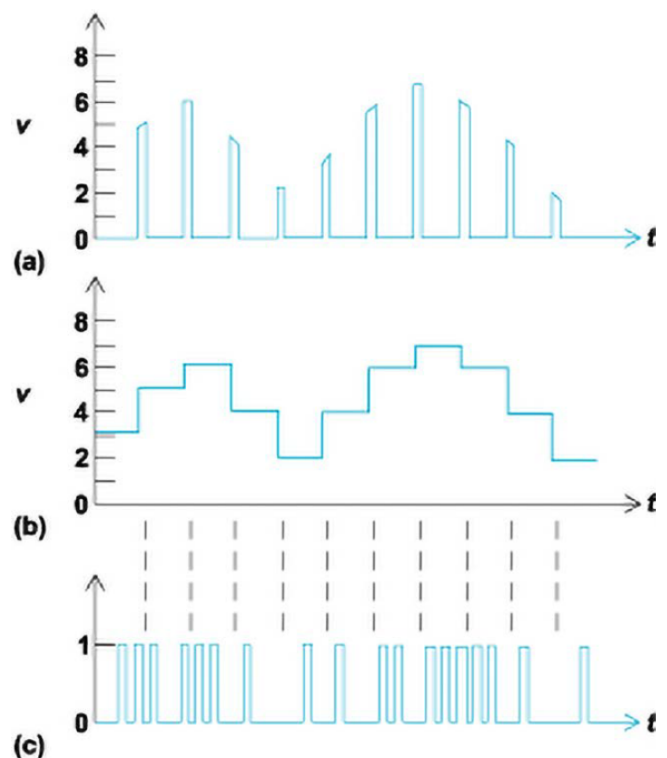
jednostce czasu. Pod tym względem więcej oferują przetworniki SAR (do 10 Msamples/s) niż przetworniki Sigma-Delta (do 1 MSamples/s).

Warto również wspomnieć o poborze prądu przetworników A/C. Przetworniki SAR są bardziej energooszczędne niż przetworniki Sigma-Delta.

Powyższe krótkie porównanie przetworników SAR i Sigma-Delta pozwala określić jaki przetwornik sprawdzi się lepiej w jakiego rodzaju aplikacjach. Przetworniki SAR cechując się wysoką częstotliwością próbkowania i niskim poborem prądu mogą być dobrym wyborem dla aplikacji szybkiej akwizycji danych oraz dla aplikacji zasilanych z baterii. Z kolei przetworniki Sigma-Delta charakteryzując się wysoką rozdzielczością wyniku pomiaru mogą być rozważane jako rozwiązanie dla dokładnych przyrządów pomiarowych np. aplikacji medycznych.

Przetwornik A/C w STM8S001J3

Mikrokontrolery STM8S dysponują dwoma przetwornikami A/C: ADC1 i ADC2. Główne ich cechy to: architektura SAR, 10-bitowa



Rysunek 1. Kolejne etapy konwersji analogowo-cyfrowej: próbkowanie (a), kwantyzacja (b) i kodowanie (c)

Tabela 1. Lista portów mikrokontrolera STM8S001J3, które mogą zostać użyte przez przetwornik ADC1

Numer pinu na obudowie S08	Nazwa pinu	Funkcja alternatywna	Funkcja alternatywna „remap”
1	PD6/AIN6/UART1_RX	Analog input 6	
6	PB4/I2C_SCL/ADC_ETR		ADC External Trigger
7	PC4/CLK_CCO/TIM1_CH4/AIN2/TIM1_CH2N	Analog input 2 ADC External Trigger	
8	PD3/AIN4/TIM2_CH2/ADC_ETR	Analog input 4	
	PD5/AIN5/UART1_TX	Analog input 5	

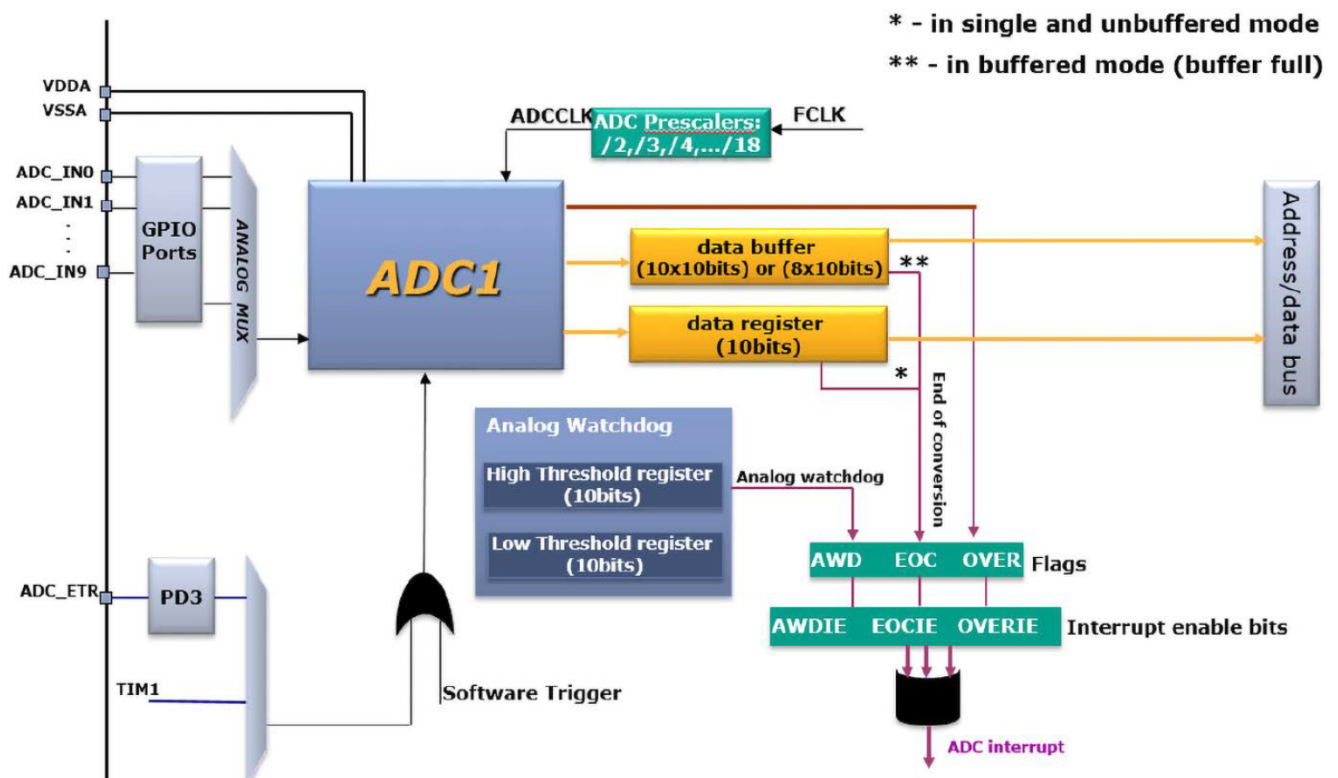
rozdzielczość wyniku pomiaru, do 16 kanałów pomiarowych (w zależności od modelu mikrokontrolera), minimalny czas pojedynczej konwersji 2,33 μ s (dla fADC max=6 MHz), typowy pobór prądu przetwornika podczas konwersji ok. 1 mA.

Schemat blokowy przetwornika ADC1 pokazano na **rysunku 2**. Wyróżnić w nim można następujące bloki:

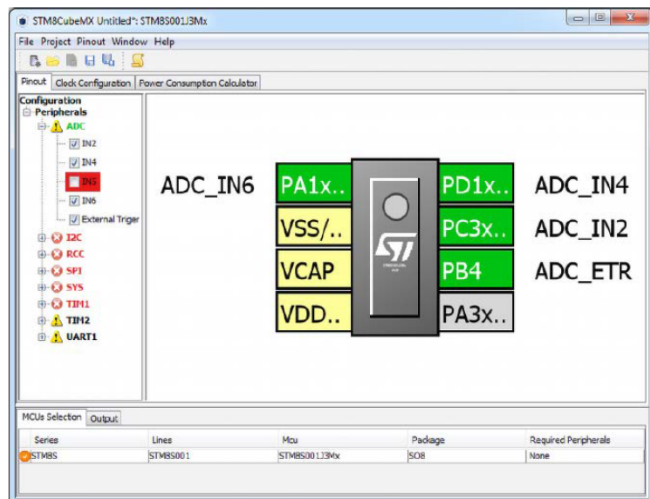
- Wyprowadzenia mikrokontrolera:
 - VDDA/VSSA odpowiedzialne za zasilanie przetwornika A/C.
 - ADC_INx będące kanałami analogowymi doprowadzającymi sygnał analogowy do przetwornika w celu jego konwersji do postaci cyfrowej.

Tabela 2. Zestawienie wybranych funkcji z biblioteki dla przetwornika A/C

Nazwa funkcji	Opis działania funkcji
ADC1_DeInit (void)	Konfiguracja domyślna przetwornika A/C
ADC1_Init (ADC1_ConvMode_TypeDef ADC1_ConversionMode, ADC1_Channel_TypeDef ADC1_Channel, ADC1_PresSel_TypeDef ADC1_PrescalerSelection, ADC1_ExtTrig_TypeDef ADC1_ExtTrigger, FunctionalState ADC1_ExtTriggerState, ADC1_Align_TypeDef ADC1_Align, ADC1_SchmittTrigg_TypeDef ADC1_SchmittTriggerChannel, FunctionalState ADC1_SchmittTriggerState)	Konfiguracja przetwornika A/C według ustawień użytkownika
ADC1_Cmd (FunctionalState NewState)	Włączenie lub wyłączenie przetwornika A/C
ADC1_DataBufferCmd (FunctionalState NewState)	Włączenie lub wyłączenie trybu buforowego, który zamiast przechowywać ostatni wynik konwersji przechowuje osiem lub dziesięć ostatnich wyników konwersji
ADC1_GetBufferValue (uint8_t Buffer)	Odczytanie wybranego wyniku konwersji z 8- lub 10-elementowego bufora
ADC1_ITConfig (ADC1_IT_TypeDef ADC1_IT, FunctionalState NewState)	Włączenie lub wyłączenie wybranych przerwań dla przetwornika A/C
ADC1_PrescalerConfig (ADC1_PresSel_TypeDef ADC1_Prescaler)	Konfiguracja podzielnika sygnału zegarowego dla przetwornika A/C
ADC1_ConversionConfig (ADC1_ConvMode_TypeDef ADC1_ConversionMode, ADC1_Channel_TypeDef ADC1_Channel, ADC1_Align_TypeDef ADC1_Align)	Konfiguracja trybu konwersji przetwornika A/C
ADC1_StartConversion (void)	Rozpoczęcie konwersji (wyzwolenie programowe)
ADC1_GetConversionValue (void)	Odczytanie wyniku konwersji
ADC1_GetFlagStatus (ADC1_Flag_TypeDef Flag)	Odczytanie stanu wybranej flagi przetwornika A/C
ADC1_ClearFlag (ADC1_Flag_TypeDef Flag)	Wyzerowanie wybranej flagi przetwornika A/C
ADC1_GetITStatus (ADC1_IT_TypeDef ITPendingBit)	Odczytanie stanu wybranego przerwania przetwornika A/C
ADC1_ClearITPendingBit (ADC1_IT_TypeDef ITPendingBit)	Wyzerowanie wybranego przerwania przetwornika A/C



Rysunek 2. Schemat blokowy przetwornika A/C (ADC1) w mikrokontrolerze STM8



Rysunek 3. Przykładowa konfiguracja mikrokontrolera STM8S001J3 w programie STM8CubeMX

- ADC_ETR, które może służyć do wyzwolenia przetwornika (rozpoczęcia konwersji) za pomocą sygnału spoza mikrokontrolera.
- Multiplexer wybierający spośród wszystkich dostępnych kanałów analogowych jeden, który w danym momencie jest połączony z przetwornikiem A/C.
- Moduł wykonujący przetworzenie (próbkiwanie, kwantowanie i kodowanie) sygnału analogowego do postaci 10-bitowego słowa cyfrowego.
- Blok wyzwalający przetwornika (źródłem wyzwolenia może być sygnał zewnętrzny na wyprowadzeniu ADC_ETR, licznik TIM1 lub kod aplikacji tzw. start programowy).
- Analog Watchdog pozwalający określić minimalny i maksymalny poziom sygnału, którego przekroczenie wygeneruje przerwanie.
- Blok przerwań:
 - AWDIE: przerwanie z Analog Watchdoga sygnalizujące przekroczenie jednego ze zdefiniowanych poziomów sygnału.

- EOCIE: przerwanie z bufora danych sygnalizujące ukończenie konwersji analogowo-cyfrowej i wpisanie jej wyniku do rejestru danych lub bufora danych.
- OVERIE: przerwanie z przetwornika A/C sygnalizujące, że ukończona została konwersja analogowo-cyfrowa, ale wynik poprzedniej konwersji nie został jeszcze odczytany z bufora danych.
- Podzielnik sygnału zegarowego (sygnał taktujący przetwornik A/C to sygnał FCLK, nazywany też fMaster, który może być podzielony przez wartość 2...18).
- Blok przechowujący wynik konwersji analogowo-cyfrowej
 - Rejestr danych przechowujący ostatni wynik konwersji analogowo-cyfrowej.
 - Bufor danych przechowujący 8 lub 10 ostatnich konwersji analogowo-cyfrowej.

W mikrokontrolerze STM8S001J3 dostępny jest jeden przetwornik – ADC1. W tabeli 1 przedstawiono zestawienie wszystkich portów mikrokontrolera STM8S001J3, które mogą zostać wykorzystane jako kanały tego przetwornika. Podsumowaniem tej tabeli może być stwierdzenie, że układ STM8S001J3 dysponuje wyprowadzeniem ADC External Trigger (które może być dostępne na pinie numer 6 lub 7) oraz czterema kanałami analogowymi (dostępnymi na pinach 1, 7 oraz 8).

Przetwornik A/C w STM8CubeMX

Narzędziem ułatwiającym pracę z portami wejścia/wyjścia mikrokontrolera STM8S001J3 jest program komputerowy STM8CubeMX. Dzięki niemu programista może w prosty sposób (za pomocą graficznego interfejsu użytkownika) sprawdzić jakie są możliwe konfiguracje dla wszystkich portów oraz peryferiów mikrokontrolera. Tak jest również w przypadku przetwornika A/C. Przykładowy scenariusz pokazano na rysunku 3.

Funkcje SPL do sterowania przetwornikiem A/C

Aby w prosty sposób skonfigurować przetwornik A/C układu STM8S001J3, a następnie realizować konwersję analogowo-cyfrową, warto w aplikacji użyć bibliotek SPL (Standard Peripheral Library) przygotowanych dla mikrokontrolerów z rodziny STM8S. Pliki

```
Listing 1. Stworzony w oparciu o funkcje SPL kod pliku main.c
#include "stm8s.h"

uint16_t ADC_value = 0;

void delay(unsigned long int how_long);

main()
{
    //delay to avoid irreversible blocking of SWIM
    delay(100000);
    //configuration of unused pins
    GPIOA->DDR |= GPIO_PIN_2;
    GPIOB->DDR |= GPIO_PIN_0 | GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3 | GPIO_PIN_6 | GPIO_PIN_7;
    GPIOC->DDR |= GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_7;
    GPIOD->DDR |= GPIO_PIN_0 | GPIO_PIN_2 | GPIO_PIN_4 | GPIO_PIN_7;
    GPIOE->DDR |= GPIO_PIN_5;
    GPIOF->DDR |= GPIO_PIN_4;
    //ADC configuration
    ADC1_DeInit();
    ADC1_Init(ADC1_CONVERSIONMODE_SINGLE, ADC1_CHANNEL_6, ADC1_PRESSEL_FCPU_D2,
    ADC1_EXTTRIG_GPIO, DISABLE, ADC1_ALIGN_RIGHT, ADC1_SCHMITTRIG_CHANNEL6, DISABLE);
    ADC1_Cmd(ENABLE);

    while (1)
    {
        ADC1_StartConversion();
        while(ADC1_GetFlagStatus(ADC1_FLAG_EOC) == RESET)
        {
        }
        ADC_value = ADC1_GetConversionValue();
        ADC1_ClearFlag(ADC1_FLAG_EOC);
    }

    void delay(unsigned long int how_long)
    {
        unsigned long int i;

        for (i = 0; i< how_long; i++)
        {
        }
    }
}
```

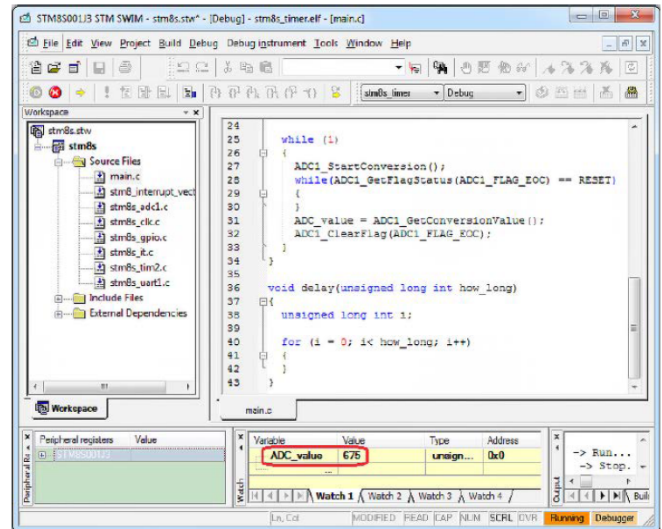
stm8s_adc1.h oraz *stm8s_adc1.c* udostępniają szereg funkcji do tego celu. Zestawienie najważniejszych funkcji zaprezentowano w tabeli 2.

Przykładowa aplikacja sterująca przetwornikiem A/C

W celu stworzenia przykładowej aplikacji użyte zostało środowisko programistyczne STVD (ST Visual Develop) oraz kompilator Cosmic CXSTM8. Opis tych narzędzi, jak również instrukcja jak stworzyć za ich pomocą szablon nowego projektu wraz z dodaniem bibliotek SPL dostępne są w artykule numer 3 z tej serii (EP 2/2018). Korzystając ze wspomnianego szablonu projektu należy edytować kod pliku *main.c*, w którym umieszczony zostanie kod aplikacji.

Przykładowa aplikacja wykorzystuje przetwornik A/C do realizacji konwersji analogowo-cyfrowej. Wybrany do tego celu został jeden kanał analogowy – ADC1_IN6. Konwersja wyzwalana jest programowo, po czym następuje oczekiwanie na flagę EOC informującą o zakończeniu konwersji. Następnie odczytywany jest wynik konwersji i wpisywany jest on do zmiennej. Na koniec zerowana jest flaga EOC. Opisane czynności powtarzane są w nieskończonej pętli. W celu zaimplementowania opisanej aplikacji wykonane zostaną następujące kroki w pliku *main.c*:

- Stworzenie zmiennej *ADC_value*, która użyta zostanie do przechowywania wyniku konwersji przetwornika A/C.
- Utworzenie funkcji opóźniającej *delay()*.
- Wywołanie funkcji opóźniającej *delay()*, co przeciwdziała przez kilka sekund ewentualnemu późniejszemu wyłączeniu interfejsu programowania i debugowania SWIM będącemu efektem rekonfiguracji portów.
- Wykonanie kodu konfiguracji portów wejścia/wyjścia, które nie są połączone z wyprowadzeniami mikrokontrolera (kod wzięty z noty aplikacyjnej AN5047: *Getting started with the STM8S001J3 microcontroller*).
- Wywołanie funkcji *ADC1_Deinit()* w celu wykonania konfiguracji domyślnej przetwornika A/C.
- Wywołanie funkcji *ADC1_Init()* w celu wykonania konfiguracji przetwornika A/C zgodnej z parametrami wybranymi przez użytkownika.
- Wywołanie funkcji *ADC1_Cmd()* w celu włączenia przetwornika A/C.
- Wewnątrz nieskończonej pętli *while()*:
 - Wywołanie funkcji *ADC1_StartConversion()* w celu rozpoczęcia konwersji przetwornika A/C.
 - W pętli *while()* oczekiwanie aż funkcja *ADC1_GetFlagStatus()* zwróci wartość flagi EOC świadczącą o końcu konwersji.
 - Wywołanie funkcji *ADC1_GetConversionValue()* w celu odczytania wyniku konwersji. Zwracana przez funkcję wartość wpisywana jest do zmiennej *ADC_value*.



Rysunek 4. Sesja debugowania środowiska STVD z podglądem „na żywo” wartości zmiennej przechowującej wynik konwersji przetwornika A/C

- Wywołanie funkcji *ADC1_ClearFlag()* w celu wyzerowania flagi EOC.

Kod zgodny z zaprezentowanym opisem pokazano w listingu 1. Działanie aplikacji obserwować można w sesji debugowania środowiska STVD. Po dodaniu zmiennej przechowującej wynik konwersji do zakładki Watch wartość tej zmiennej będzie odświeżana automatycznie. Obrazuje to rysunek 4. Przykładowo odczytany wynik konwersji ma wartość 675. Wiedząc, że 10-bitowy przetwornik przyjmuje wartość maksymalną równą 1023 (2 do potęgi 10 minus 1) i że dla wartości tej napięcie na wejściu przetwornika jest równe napięciu zasilania (3,3 V), można w następujący sposób oszacować wartość napięcia dla odczytanej wartości:

$$Napięcie = \frac{675 \cdot 3,3V}{1023} = 2,17V$$

Podsumowanie

W artykule przekazano podstawowe informacje o przetworniku A/C mikrokontrolera STM8S001J3 wraz z opisem przykładowej aplikacji. Osoby chcące dowiedzieć się bardziej szczegółowych informacji powinny sięgnąć do dokumentacji technicznej producenta. Parametry i charakterystyka przetwornika A/C dostępne są w nocie katalogowej mikrokontrolera (*datasheet*). Z kolei schematy oraz opis wszystkich funkcjonalności i rejestrów znajduje się w podręczniku użytkownika mikrokontrolera (*user manual* RM0016).

Szymon Panecki
szymon.panecki@st.com

REKLAMA

NAJLEPSZY MOBILNY
ADRES W SIECI
HTTP://M.EP.COM.PL

