

---

## Ćwiczenie 6: Obsługa przetwornika analogowo-cyfrowego (ADC)

Instrukcja laboratorium

Mariusz Chilmon <[mariusz.chilmon@ctm.gdynia.pl](mailto:mariusz.chilmon@ctm.gdynia.pl)>



2024-01-20

People think that computer science is the art of geniuses but the actual reality is the opposite, just many people doing things that build on each other, like a wall of mini stones.

— Donald Knuth

## Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z:

- podstawową konfiguracją ADC,
- odczytem pomiaru z ADC,
- zmianą częstotliwości pomiarów,
- przeliczeniem rejestru pomiaru ADC na napięcie,
- przeliczeniem zmierzonego napięcia na temperaturę.

## Uruchomienie programu wyjściowego

1. Podłącz płytkę *WPSH209* do *Arduino Uno*.
2. Podłącz termometr LM35DZ do linii A5.
3. Wynik pomiaru na wyświetlaczu ma wartość 00.00.

## Zadanie podstawowe

Celem zadania podstawowego jest uruchomienie pomiaru temperatury z wykorzystaniem ADC mikrokontrolera i scalonego termometru analogowego LM35DZ.



Przetwornik analogowo-cyfrowy (ADC, ang. *Analog-to-Digital Converter*) zamienia sygnał analogowy, na ogół napięcie, na wartość cyfrową. Może być zewnętrznym układem scalonym lub, jak w tym przypadku, być wbudowany w mikrokontroler.

## Wymagania funkcjonalne

1. Na wyświetlaczu prezentowana jest temperatura mierzona przez LM35DZ w °C z rozdzielczością do części setnych.

Pomiar, szczególnie na pozycjach mniej znaczących, może zmieniać się bardzo szybko przez co nie zawsze będzie czytelny.

## Modyfikacja programu

### Konfiguracja multiplekserów i preskalera

Zastosowany w mikrokontrolerze przetwornik analogowo-cyfrowy sukcesywnej aproksymacji wymaga doprowadzenia do przetwornika cyfrowo-analogowego (DAC) napięcia referencyjnego, względem którego będzie wykonywany pomiar (obwód zielony). W tym przypadku wykorzystujemy wbudowane w mikrokontroler napięcie odniesienia 1,1 V. Wybór źródła napięcia referencyjnego odbywa się za pomocą bitów `REFS0...REFS1`.

Mierzone napięcie jest z kolei doprowadzane do komparatora próbkującego (obwód czerwony). Wykorzystujemy wejście `ADC5`, do którego podłączony jest termometr LM35DZ. Wybór mierzonego sygnału odbywa się za pomocą bitów `MUX0...MUX3`.

By działanie ADC było w ogóle możliwe, musi on być taktowany odpowiednim sygnałem zegarowym. Z dokumentacji mikrokontrolera można odczytać przedział częstotliwości, w których producent gwarantuje zachowanie optymalnych parametrów pomiaru, w szczególności zachowanie 10-bitowej rozdzielczości. Uzyskanie odpowiedniej częstotliwości jest możliwe przy dowolnej częstotliwości pracy mikrokontrolera dzięki preskalerowi o odpowiednim zakresie (obwód niebieski). Wybór stopnia podziału odbywa się za pomocą bitów `ADPS0...ADPS2`.

### Uruchomienie pomiaru

Po skonfigurowaniu multiplekserów i preskalera należy:

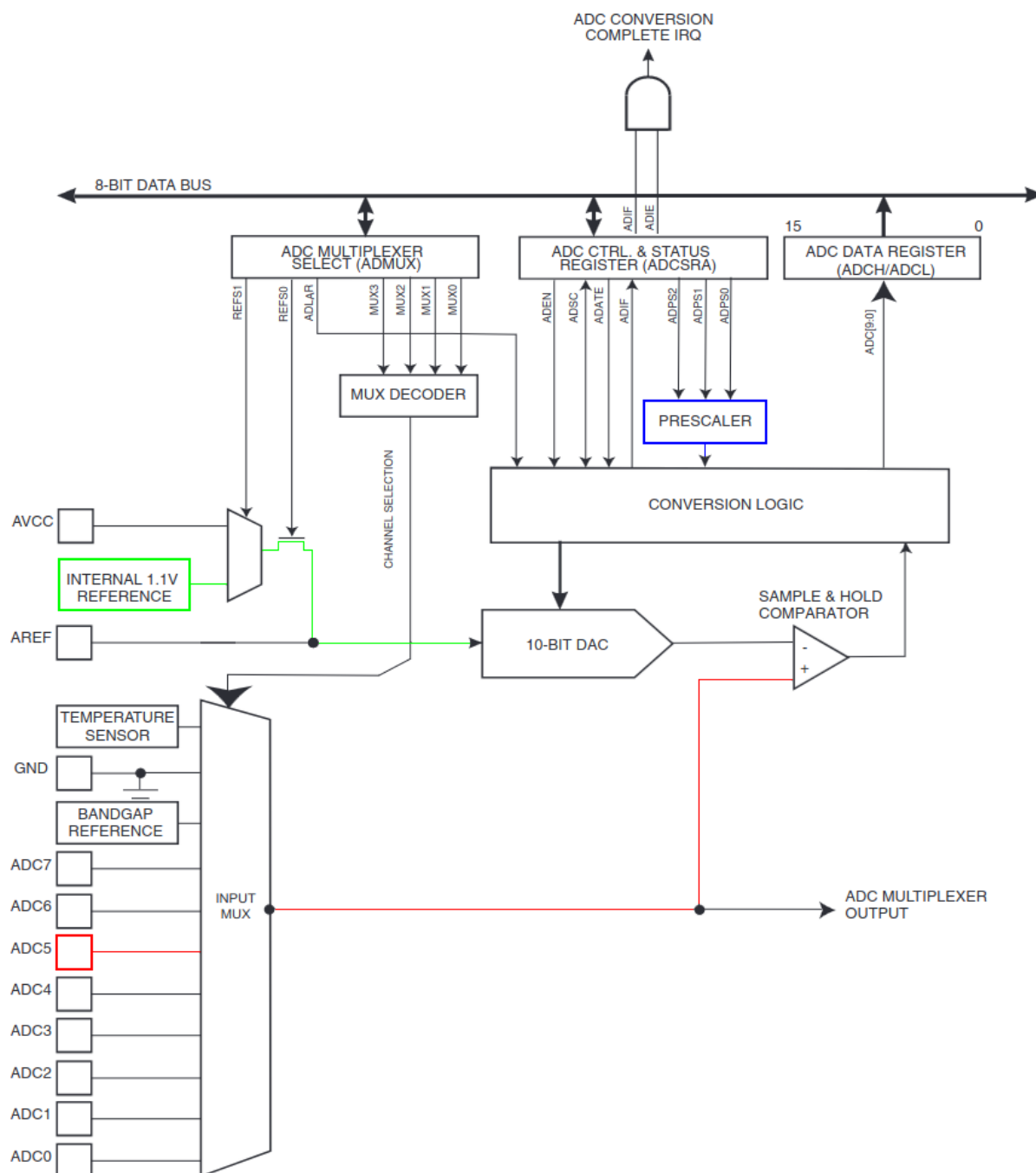
1. Włączyć tryb autowyzwalania przez ustawienie bitu `ADATE` (*ADC Auto Trigger Enable*), dzięki czemu pomiar nie będzie wykonany jednorazowo, ale po zakończeniu jednego pomiaru automatycznie będzie uruchamiany następny.
2. Włączyć ADC przez ustawienie bitu `ADEN` (*ADC Enable*).
3. Uruchomić pierwszą konwersję (pomiar) przez ustawienie bitu `ADSC` (*ADC Start Conversion*).



Wszystkie trzy powyższe bity można ustawić jednocześnie, także wraz z bitami `ADPS0...ADPS2`.



Konfigurację i uruchomienie przetwornika analogowo-cyfrowego umieść w funkcji `adcInitialize()` w pliku `main.cpp`.



Rysunek 1: Schemat blokowy ADC

## Odczyt pomiaru

Wynik pomiaru należy odczytać z rejestrów **ADCL** i **ADCH** (pomiar jest 10-bitowy, więc jego wynik mieści się w dwóch rejestrach). Ponieważ mamy do czynienia z mikroprocesorem 8-bitowym, nie jest możliwy odczyt obu rejestrów w czasie jednej instrukcji. Może więc dojść do sytuacji, gdy między odczytem jednej i drugiej części pomiaru, pomiar ulegnie zmianie, w wyniku czego odczytamy część starszego i część nowszego pomiaru. Aby uniknąć takiej sytuacji, producent gwarantuje, że po odczytaniu rejestru **ADCL** zawartość **ADCH** nie ulegnie zmianie, dopóki nie zostanie odczytana. Należy więc zawsze w pierwszej kolejności odczytywać rejestr **ADCL**<sup>1</sup>.

Odczytaną wartość należy przeliczyć na napięcie, a następnie na temperaturę.



Odczyt i przeliczenie pomiaru umieść w funkcji `measure()` (plik `main.cpp`), która jest wywoływana przy każdym odświeżeniu wyświetlacza, a zwracana przez nią wartość jest na nim umieszczana.

## Zadanie rozszerzone

Celem zadania rozszerzonego jest ułatwienie odczytu przez ograniczenie szybkości pomiaru.

### Wymagania funkcjonalne

1. Pomiar następuje nie częściej niż kilka razy na sekundę.

### Modyfikacja programu

Ustal odpowiedni dzielnik taktujący *Timer/Counter1* (bity **CS10...CS12**). Flagą **TOIE1** włącz przerwanie od przepełnienia tego timera. Użyj bitów **ADTS0...ADTS2**, aby wybrać taktowanie pomiaru tym przerwaniem (*Timer/Counter1 Overflow*).

Częstotliwość przerwania w zależności od wybranego dzielnika  $N$  dana jest wzorem:

$$f = \frac{f_{clkI/O}}{2^{16}N} \quad (1)$$

---

<sup>1</sup>Kompilator AVR-GCC ma zdefiniowaną zmienną **ADC**, która umożliwia dostęp do obu rejestrów jako 16-bitowej całości, jednak jego dokumentacja nigdzie nie gwarantuje poprawnej kolejności odczytu.