

2018

PDF wersja 1.0

ATNEL

Mirosław Kardaś



## ATB ANALOG INDICATOR PLUS AVR GCC LIBRARY

# [ INSTRUKCJA ATB-AI ]

Moduł ATB-ANALOG INDICATOR + biblioteka w języku C do obsługi rejestrów przesuwanych, expanderów I2C oraz przetwornika ADC wraz z precyzyjnym pomiarem napięcia z dokładnością 10mv oraz możliwością uśredniania pomiarów (cyfrowe filtry dolnoprzepustowe)

## Spis treści

Podstawowe informacje.....	2
Konfiguracje sprzętowe modułu.....	3
Właściwości biblioteki C.....	6
Pozostałe ustawienia. ....	7
Funkcje zwrotne – Callbacks modułu AI. ....	8
Pierwsza funkcja callback. ....	8
Druga funkcja callback. ....	8
Trzy opcje do błyskawicznej zmiany podstawowych ustawień.....	10
Schemat modułu .....	11
Polecana literatura .....	12
INFORMACJE KONTAKTOWE .....	13

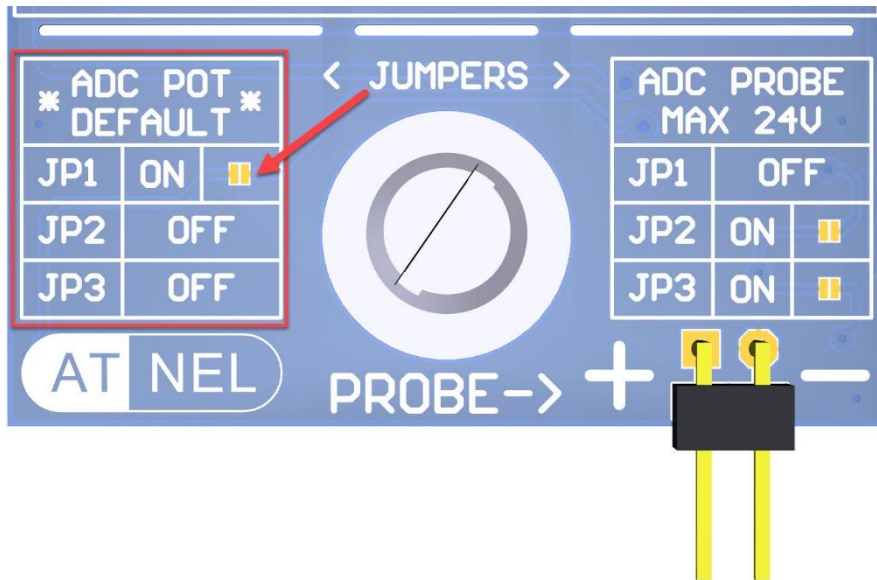
## Podstawowe informacje

**Moduł ATB ANALOG INDICATOR** może spełniać dwie różne funkcje. Pierwsza funkcja to analogowy sterownik (*manipulator/regulator zadanej wartości w konkretnym zakresie*) oparty o potencjometr obrotowy 22K. Druga funkcjonalność umożliwia wygodne pomiary napięcia w oparciu o mikrokontroler oraz kalibrację budowanego miernika. Działanie modułu oparte jest o przetwornik **ADC** w mikrokontrolerze. Moduł może działać w oparciu o magistralę **SPI** lub **I2C**. Na magistrali SPI zastosowano układ 16-bitowego rejestru przesuwającego **SCT2024**, nie mniej jednak biblioteka pozwala używać zamiennie np. dwóch 8-bitowych rejestrów przesuwających typu **74HC595**. Dla magistrali I2C przewidziano obsługę w oparciu o 16-bitowy ekspander **PCF8587**.

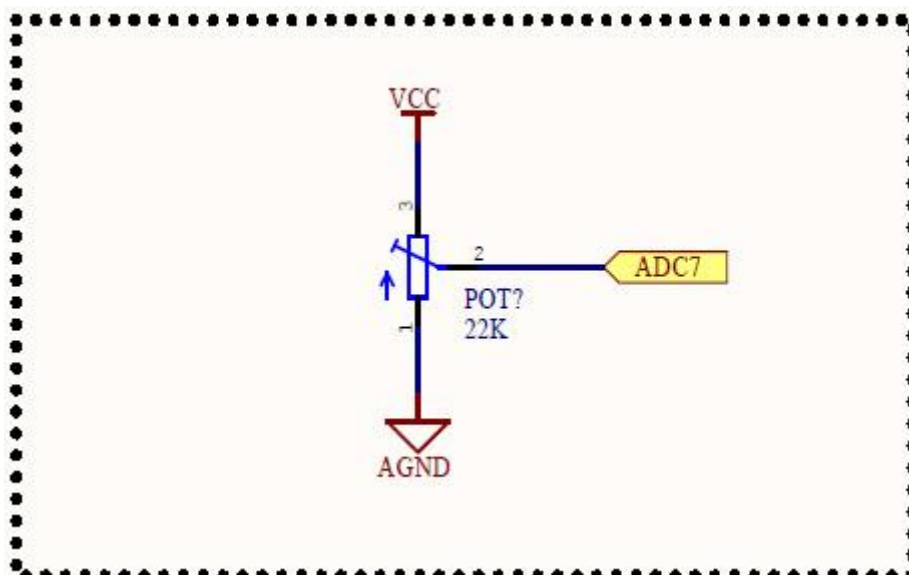
- ❖ Zasilanie układu +5 V.
- ❖ Do wyboru moduły albo na magistralę SPI (SCT2024) albo na magistralę I2C (PCF8575).
- ❖ W pełnej (zlutowanej) wersji modułu – 16 diod LED.
- ❖ Możliwość sterowania jasnością diod dla wersji z magistralą SPI.
- ❖ Możliwość wyboru jednej z dwóch funkcjonalności za pomocą zworek SMD.
- ❖ Wyprowadzone goldpiny do podłączenia sond służących do pomiarów napięć.
- ❖ Wizualizacja mierzonego napięcia za pomocą diod LED.
- ❖ Wizualizacja pozycji potencjometru w trybie manipulatora.
- ❖ Tryb regulatora umożliwia sterowanie dowolnym zakresem wartości (zgodnych z typem int).
- ❖ Możliwość odłączenia (odłamania) regulatora z potencjometrem i pozostawienie samego wyświetlacza analogowego na 16 diodach LED.
- ❖ Przygotowana rozbudowana biblioteka w języku C do obsługi wszystkich wariantów pracy.
- ❖ Możliwość korzystania z 8-bitowego i 11-bitowego trybu ADC.
- ❖ Możliwość pracy biblioteki i wyświetlania wyników w tzw. trybie „serwo”.
- ❖ Moduł wykonany w standardzie ATB-MICROPORT, dzięki czemu można go najwygodniej stosować w zestawach uruchomieniowych **ATB w wersjach tylko 1.05a (w obecnej wersji modułu nie można go stosować w starszych wersjach zestawów 1.05)**, ale także na płytkach stykowych.
- ❖ **Przy zakupie pełnego (zlutowanego) modułu biblioteka C dodawana jest gratis.**

## Konfiguracje sprzętowe modułu

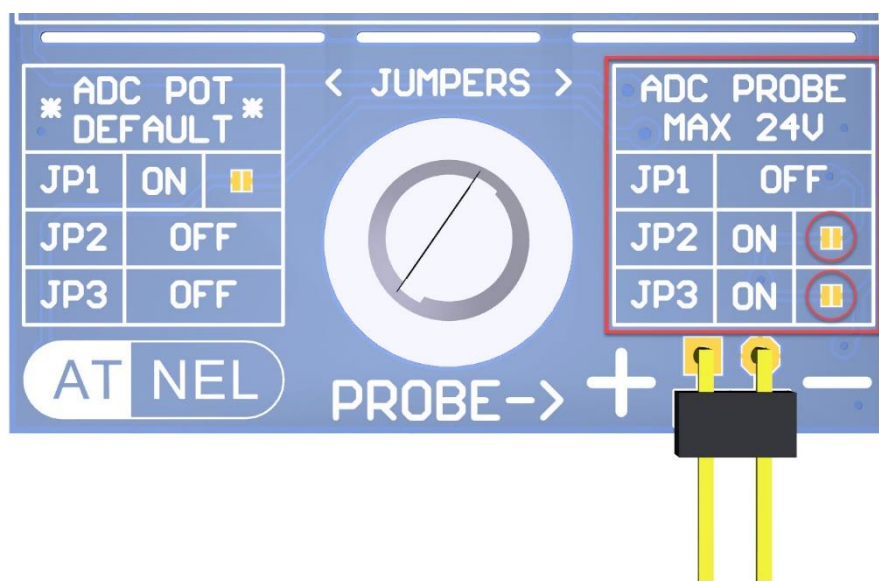
1. Praca w trybie manipulatora/regulatora przy pomocy potencjometru obrotowego o wartości 22K, wymaga zwarcia zworki SMD JP1 oraz rozwarcia zworek SMD JP2 i JP3, zgodnie z obrazkiem poniżej.



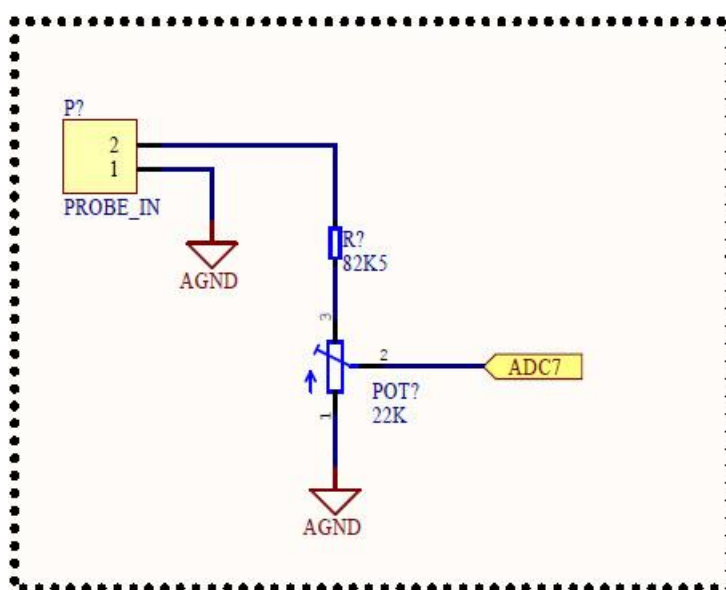
Praca w tym trybie oparta jest o proste podłączenie potencjometru pomiędzy VCC i GND, zaś suwak doprowadzony jest do wejścia mikrokontrolera (PA7) ADC7, zgodnie ze schematem poniżej:



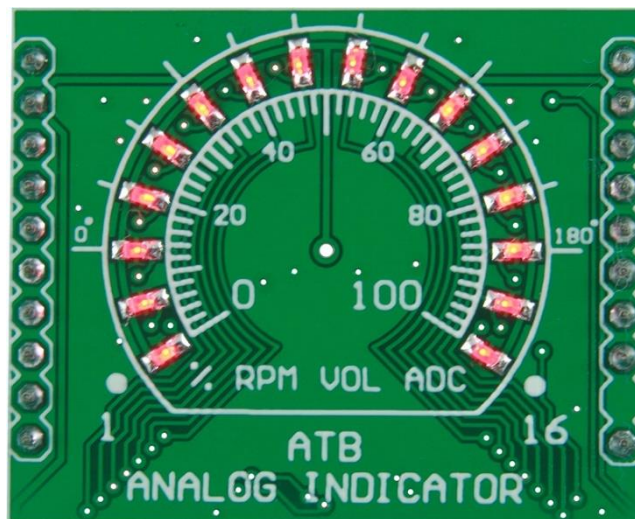
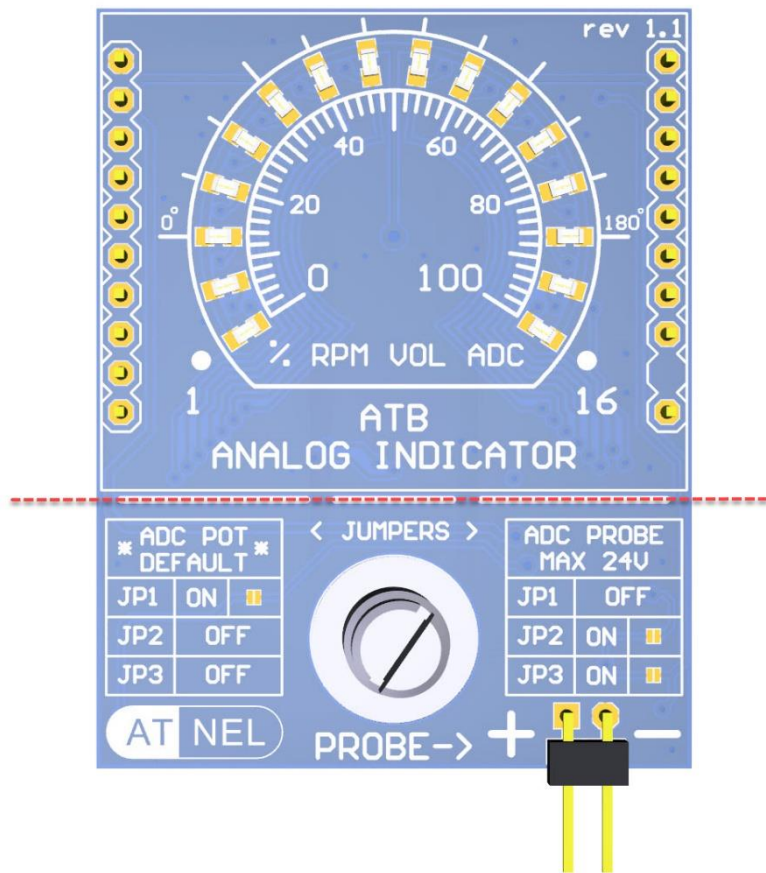
- Praca w trybie **miernika napięcia stałego** z maksymalnym zakresem 24 V, wymaga rozwarcia zworki **JP1** oraz zwarcia dwóch zworek **JP2** i **JP3**.



Praca w tym trybie oparta jest o podłączenie potencjometru jako dolnego rezystora w dzielniku rezystorowym, na który podawane jest mierzone napięcie. Sygnał napięciowy z suwaka potencjometru podawany jest na wejście mikrokontrolera (PA7) ADC7, zgodnie ze schematem poniżej. Z uwagi na wartość rezystancji górnego rezystora 82K5 oraz maksymalną wartość rezystancji potencjometru obrotowego 22K, układ pozwala w bezpieczny sposób mierzyć maksymalne napięcie +24 V. Przy takim napięciu i maksymalnym przesunięciu suwaka w kierunku górnego rezystora 82K5, na wyjściu dzielnika odłoży się maksymalnie ok +5,5 V. W przypadku konieczności pomiarów większych wartości napięć należy we własnym zakresie zwiększyć wartość górnego rezystora.



3. **Praca w trybie wyświetlacza linijki LED.** Należy odłamać wzdłuż frezowanych podłużnych otworów część płytki PCB zawierającej potencjometr ze zworkami SMD. Na rysunku poniżej miejsce przełamania zaznaczono czerwoną przerywaną linią.



Widok modułu w rzeczywistości po odłamaniu części manipulatora.

## Właściwości biblioteki C

Wszystkie parametry konfiguracyjne biblioteki `Mk_AANALOG_INDI`, dostępne są w pliku nagłówkowym `mk_analog_indi.h`. Szczegółowe parametry konfiguracyjne znajdują się w sekcji:

```
/*----- Library configuration -----*/
```

Na samym początku należy określić magistralę na której ma pracować moduł:

```
#define BUS_TYPE 0 // 0-SPI (SCT2024), 1-I2C (PCF8574), 2-none
```

w komentarzu po prawej stronie podano wartości parametrów. W kolejnym kroku należy bezwzględnie ustawić piny dla portu programowego SPI lub adresu układu na magistrali I2C, w sekcjach:

```
/****** soft SPI CONFIG *****/
#define MOSI_PORT PORTB
#define SCK_PORT PORTB
#define LT_PORT PORTB

#define MOSI_DIR DDRB
#define SCK_DIR DDRB
#define LT_DIR DDRB

#define LT (1<<PB4)
#define MOSI (1<<PB5)
#define SCK (1<<PB7)

/****** LED count - config *****/
#define LED_COUNT 16 // LEDs count for SPI/I2C

/****** i2c address - config *****/
#define PCF8574_ADR 0x4E
```

W przypadku magistrali SPI należy pamiętać, że komunikacja SPI realizowana jest programowo a zatem można użyć dowolnych pinów mikrokontrolera na dowolnym porcie. Pomędzy sekcjami SPI/I2C umiejscowiono parametr „`LED_COUNT`”, za pomocą którego ustalamy liczbę diod w linijce LED. Naturalnie biblioteka może obsłużyć zarówno mniej niż 16 jak i więcej niż 16 we własnych realizacjach modułu.

Konfiguracja przewiduje również wykorzystanie biblioteki nawet bez użycia modułu. Biblioteka może pracować tylko w oparciu o prostą linijkę ośmiu diod LED podłączonych do wybranego jednego pełnego portu mikrokontrolera. Należy oczywiście skonfigurować wybrany port w sekcji:

```
/****** External uC PORT - config *****/
#define USE_OUT_PORT 0 // 1-parallel 8-bit uC PORT

#define PLED_COUNT 8 // LEDs count for 8-bit parallel port
#define USE_REVERSE_BITS_IN_BYTE 1 // invert bits in byte
#define OUT_PORT PORTC
#define OUT_DIR DDRC
```

## Pozostałe ustawienia.

Możliwość inwersji diod LED w zależności od rodzaju inwersji sygnałów w konkretnym zastosowanym rejestrze przesuwającym, ekspanderze i2c lub po prostu ze względu na podłączenie diod katodą lub anodą do pinów sterujących układem. Domyślnie dla rejestrów SCT2024 wartość parametru = ZERO.

```
#define INVERT_LEDS 0
```

Z uwagi na możliwość zastosowania modułu do sterowania serwomechanizmami, wprowadzono tzw. tryb „**servo mode**”, który pozwala sterować za pomocą potencjometru całym zakresem wartości podanym przy inicjalizacji biblioteki a jednocześnie wyświetlacz diod LED będzie uwzględniał tylko te, które mieszczą się w zakresie 0° do 180°. Pomijane są zatem cztery diody. Dwie na początku i dwie na końcu linijki.

```
#define SERVO_MODE 0 // 0-servo mode OFF, 1-servo mode ON
```

Moduł przewiduje i posiada rozbudowany sposób pomiarów ADC zarówno do celów manipulatora jak również do budowy miernika napięcia. Pomiar ADC można również w ogóle wyłączyć, jeśli wyświetlacz ma być tylko sterowany cyfrowo z poziomu programu w mikrokontrolerze. Można zatem wybrać dwa dostępne w mikrokontrolerze tryby pracy ADC 8- oraz 11- bitowy.

```
#define USE_ADC 1 // 0-no ADC, 1-ADC 8-bit, 2-ADC 10-bit
```

Z uwagi na zakłócenia podczas pomiaru napięcia wprowadzono opcję uśredniania pomiarów wartości ADC niezależnie od tego czy są one dokonywane w trybie 8- czy 11- bitowym. Uśrednianie wartości pomiarów ADC dokonujemy poprzez ustawienie wartości = JEDEN dla parametru o nazwie „**USE\_SMOOTH\_ADC**”. Dodatkowo wyprowadzono dolnoprzepustowy filtr cyfrowy, w którym można ustalać ilości próbek do uśredniania, co powoduje naturalnie zmianę czasu uśredniania przy różnych wartościach. Parametr odpowiedzialny za filtr dolnoprzepustowy to: „**SMOOTH\_ADC\_FACTOR**”.

```
#define USE_SMOOTH_ADC 0 // 0-disabled, 1-enabled
#define SMOOTH_ADC_FACTOR 16 // minimum - 4, maximum - 256
```

Decydując się na wykorzystanie ADC należy ustawić parametr o nazwie „**USE\_VREF\_TYPE**”. Tuż ponad wyborem opcji zdefiniowane są trzy stałe o nazwach: „**REFS\_VCC**”, „**REFS\_2\_56**” oraz „**REFS\_1\_1**”. **Uwaga!** Na tym etapie należy podać ustawienia bitów o nazwie **REFSx** (umiejscowionych w rejestrze ADMUX) dla przedstawionych poniżej wartości, na podstawie noty PDF własnego mikrokontrolera. Domyślnie podano przykłady dla mikrokontrolera ATmega32 i dlatego nie ma przypisanej wartości dla „**REFS\_1\_1**”, ponieważ ten mikrokontroler nie posiada wbudowanego napięcia odniesienia 1,1 V. Ostatecznie jedną z trzech zdefiniowanych wartości należy przypisać do parametru „**USE\_VREF\_TYPE**”.

```
//----- set these values REFSx for your uC -----
#define REFS_VCC (1<<REFS0) // AREF = VCC
#define REFS_2_56 (1<<REFS1) | (1<<REFS0) // AREF = 2,56 V
#define REFS_1_10 (1<<REFSx . . . // AREF = 1,1 V

#define USE_VREF_TYPE REFS_VCC // set desired value
```



## Funkcje zwrotne – Callbacks modułu AI.

Biblioteka umożliwia wykorzystanie dwóch funkcji zwrotnych typu Callback.

### Pierwsza funkcja callback.

Jeśli zostanie aktywowana zapewnia przekazywanie do programu głównego danych podczas zmieniających się parametrów takich jak:

1. Numer diody LED.
2. Wartość procentową wybranego zakresu za pomocą potencjometru.
3. Wartość całkowitą z zadanego zakresu.

Aby zarejestrować własną funkcję należy użyć funkcji rejestrującej o nazwie: `register_ai_callback()`. Odpowiada za to część sekcji:

```
/*----- CALLBACK's - config -----*/
#define USE_AI_CALLBACK      0      // 0-disable, 1-enable
```

### Druga funkcja callback.

Zapewnia przede wszystkim pomiar napięcia. Zwraca użytkownikowi do programu głównego trzy argumenty:

1. Wartość ADC.
2. Wartość mierzonego napięcia w mili woltach z dokładnością do dwóch miejsc po przecinku.
3. String reprezentujący wartość napięcia z dokładnością do dwóch miejsc po przecinku.

Aby zarejestrować własną funkcję odczytu mierzonego napięcia, należy użyć funkcji rejestrującej o nazwie: `register_ai_vcallback()`. Odpowiada za to część sekcji:

```
/*----- CALLBACK's - config -----*/

//--- voltage masurement callback and properties -----
#define USE_VOLTAGE_CALLBACK      0      // 0-disable, 1-enable
#define SMOOTH_VOLTAGE_FACTOR    32     // minimum - 4, maximum - 256
#define ADC_ERROR_CORRECTION     1     // 0-OFF, 1-ON

#if USE_ADC == 2 // wartości średniej wartości napięcia i ADC 10-bit
    #define MID_VOLT      12080      // middle Voltage*1000
    #define MID_ADC      498
#endif

#if USE_ADC == 1 // wartości średniej wartości napięcia i ADC 8-bit
    #define MID_VOLT      5000      // middle Voltage*1000
    #define MID_ADC      105      //
#endif
/*----- CALLBACK's - config END -----*/
```

Druga funkcja jest znacznie bardziej rozbudowana niż pierwsza, dlatego występują dodatkowe właściwości/parametry do ustawienia na potrzeby jej działania. Parametr „SMOOTH\_VOLTAGE\_FACTOR” jak można się domyślić zapewnia uśrednianie za pomocą cyfrowego filtra dolnoprzepustowego mierzonej wartości napięcia. Wartość tego parametru w rozpiętości od 2 do 256 pozwala na szeroki zakres regulacji i dopasowanie czasu uśredniania do własnych potrzeb. Warto eksperymentować z różnymi wartościami tego parametru przed ostatecznym wdrożeniem.

Kolejny parametr „ADC\_ERROR\_CORRECTION” zapewnia (mocno) uproszczoną korekcję błędów pomiarów ADC opisanych w nocie mikrokontrolera AVR, takich jak: błędy wzmacnienia, błędy nieliniowości itp.

Bodajże najważniejsze parametry związane z nietypowym podejściem do pomiarów napięcia za pomocą modułów ADC określone są w sekcjach:

1. `#if USE_ADC == 2`
2. `#if USE_ADC == 1`

W zależności od wybranej rozdzielczości ADC (8- czy 11- bit) należy dokonać pomiaru środkowego zakresu maksymalnego docelowego mierzonego napięcia a także wartości ADC odpowiadającej mierzonej wartości napięcia. Należy pamiętać aby zmierzoną wartość napięcia pomnożyć x1000 podobnie jak wartości odpowiadającej wartości ADC.

Dzięki zaprezentowanemu tutaj podejściu do pomiaru napięcia można w 100% pominąć błędy związane z takimi parametrami jak:

1. Procentowa wartość tolerancji rezystorów użytych w dzielniku rezystorowym.
2. Błędu związanego z odchyłką wartości wbudowanego napięcia odniesienia w mikrokontrolerze AVR.

Zasada doboru parametrów: „MID\_VOLT” oraz „MID\_ADC”. Na początek należy zdecydować jaki będzie maksymalny zakres mierzonego napięcia. Dla modułu ATB-ANALOG INDICATOR, z uwagi na dzielnik napięcia składający się z rezystorów: 82,5K oraz potencjometru maks 22K, napięcie maksymalne może wynosić +24V. Jeśli konieczny byłby większy zakres napięcia, należałoby zwiększyć wartość rezystora 82,5K na większą.

Przy założeniu np. maksymalnej mierzonej wartości +24V, należy przyłożyć napięcie 24-25V do zacisków PROBE. Następnie za pomocą potencjometru doprowadzić do skrajności wartości ADC 1022-1023. Gdy uzyskana zostanie skrajna wartość 1023. Należy obniżyć napięcie do połowy maksymalnej wartości i przy pomocy demonstracyjnego programu ustawić wartości parametrów: „MID\_VOLT” oraz „MID\_ADC” w zależności od wybranej rozdzielczości 8- lub 11- bit dla ADC. To wszystko. Biblioteka przy udziale uproszczonej korekcji błędów nieliniowości przetworników ADC dla mikrokontrolerów AVR powinna zapewnić płynny i pewny odczyt mierzonego napięcia w porównaniu do sprzętowego miernika, który został użyty do pomiarów wzorcowych.

Użytkownik może we własnym zakresie zmienić sposób pomiaru napięcia we własnym zakresie, odpowiednio modyfikując funkcje biblioteczne.

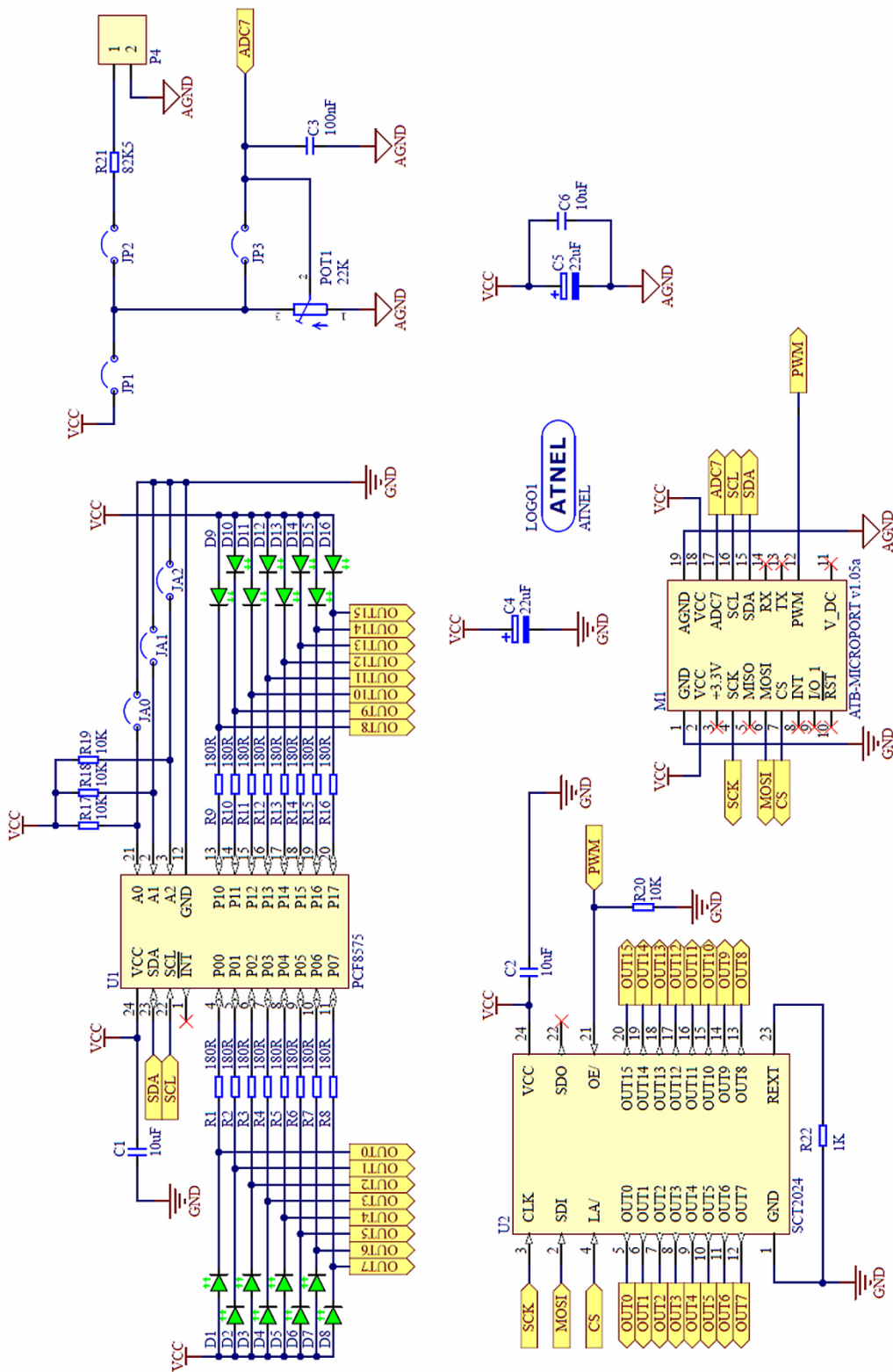
## Trzy opcje do błyskawicznej zmiany podstawowych ustawień.

Z uwagi na dość dużą ilość parametrów na początku pliku przygotowano trzy makra, które domyślnie są zakomentowane, a pozwalają na dokonanie szybkiej zmiany trzech podstawowych funkcjonalności biblioteki. Wszystko znajduje się w sekcji:

```
/*----- uncomment one from three lines below -----*/  
  
I // #define POT // QUICK configure for POTENTIOMETER CONTROL  
II // #define PROBE_256 // QUICK configure for Voltage with AREF 2,56 V  
III // #define SHOW_DEMO // Enable DEMO MODE - started while init function  
  
/*----- comment three lines above to configure all manually -----*/
```

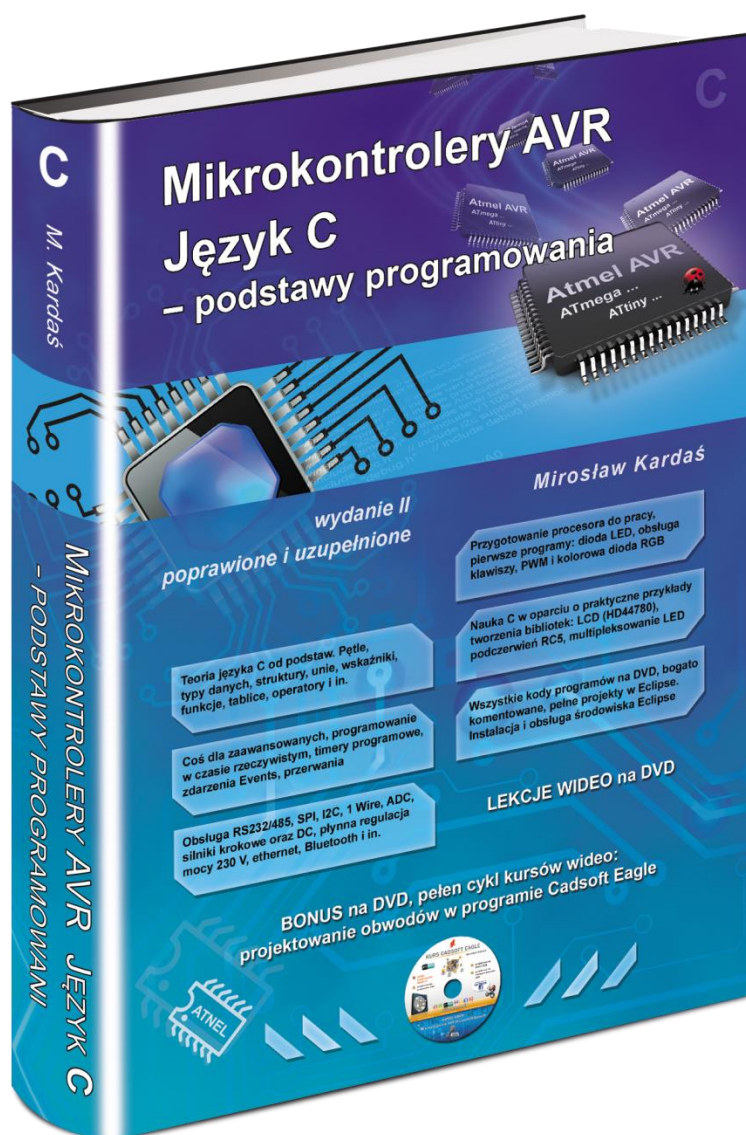
- I. **POT** – odkomentowanie pierwszej linii powoduje szybką konfigurację funkcjonalności manipulatora w postaci potencjometru obrotowego.
- II. **PROBE\_256** – odkomentowanie drugiej linii dokonuje szybkiej konfiguracji biblioteki w trybie miernika napięcia z zastosowaniem napięcia odniesienia 2,56 V. Jest to opcja przewidziana dla mikrokontrolerów typu ATmega8/16/32.
- III. **SHOW\_DEMO** – odkomentowanie trzeciej linii spowoduje uruchomienie nieskończonej pętli z trybem DEMO dla wyświetlacza opartego o 16 diod LED znajdujących się w module. Procedura uruchamiana jest podczas inicjalizacji biblioteki w głównej funkcji programu.

### Schemat modułu



## Polecana literatura

Zalecany podręcznik do nauki podstaw programowania mikrokontrolerów w języku C z zestawami uruchomieniowymi serii ATB firmy Atmel:



<http://atnel.pl/mikrokontrolery-avr-jezyk-c.html>

Zapraszamy na nasz blog: [www.mirekk36.blogspot.com](http://www.mirekk36.blogspot.com)

Zapraszamy na nasze forum wsparcia technicznego: [www.forum.atnel.pl](http://www.forum.atnel.pl)

Zapraszamy do wideo poradników na youtube: [www.youtube.com/mirekk36](http://www.youtube.com/mirekk36)

Zapraszamy na nasz fanpage na Facebook: [www.facebook.com/atnel.mikrokontrolery](http://www.facebook.com/atnel.mikrokontrolery)

## INFORMACJE KONTAKTOWE

**ATNEL Mirosław Kardaś**

**Adres:**

**ul. Kurza 24,  
70 - 795 Szczecin**

**Telefon:**

**+48 91 4635 683**

**+48 501 008 523**

**Strona Internetowa:**

[www.atnel.pl](http://www.atnel.pl)

[www.sklep.atnel.pl](http://www.sklep.atnel.pl)

**e-mail:**

[biuro@atnel.pl](mailto:biuro@atnel.pl)

[sklep@atnel.pl](mailto:sklep@atnel.pl)

