
Ćwiczenie 12: Obsługa klawiatury analogowej

Instrukcja laboratorium

Mariusz Chilmon <mariusz.chilmon@ctm.gdynia.pl>



2024-04-06

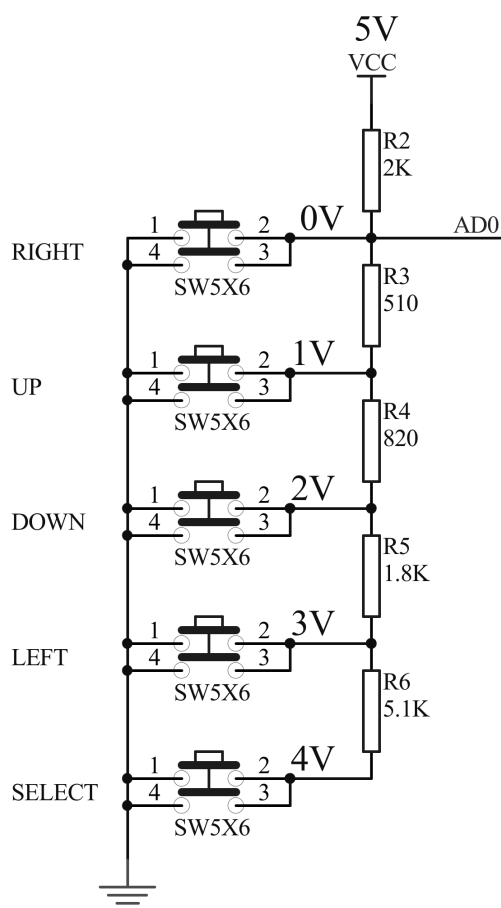
Confusion is part of programming.

— Felienne Hermans

Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z:

- działaniem klawiatury analogowej,
- konwersją liczb na łańcuch znaków za pomocą biblioteki standardowej C i C++,
- zakresem typów całkowitych.



Rysunek 1: Schemat ideowy klawiatury analogowej

Klawiatura analogowa wykorzystuje dzielnik napięciowy, sterowany za pomocą przycisków. Sygnałem wyjściowym jest napięcie, które może być zmierzone przez pojedynczy pin mikrokontrolera.

Uruchomienie programu wyjściowego

1. Podłącz płytkę *LCD Keypad Shield* do *Arduino Uno*.
2. Na wyświetlaczu widoczne są etykiety bez żadnych wartości.



```
ADC:      K:
COUNTER:
```

Rysunek 2: Wyjściowy stan wyświetlacza

Zadanie podstawowe

Celem zadania podstawowego jest uzupełnienia sterownika klawiatury, tak by rozpoznawał aktualnie wciśnięty przycisk.

Wymagania funkcjonalne

1. W polu **ADC** wyświetlana jest wartość odczytana z ADC.
2. W polu **K** wyświetlana jest nazwa wciśniętego przycisku.



```
ADC: 619 K: LEFT
COUNTER:
```

Rysunek 3: Stan wyświetlacza przy wciśniętym przycisku *LEFT* (wartość odczytana z ADC może się nieznacznie różnić)

Modyfikacja programu

Odczyt i wyświetlenie pomiaru z ADC

Metoda `Keypad::measure()` odczytuje wynik pomiaru z ADC, który w funkcji `printMeasure()` należy wypisać na wyświetlaczu w polu **ADC**. W tym celu należy posłużyć się buforem `char buf[16]` oraz funkcją `snprintf()` z biblioteki `stdio.h`:

```
1 char buf[16];
2 snprintf(buf, sizeof(buf), "%4u", keypad.measure());
3 lcdDisplay.write(buf);
```

Na tym etapie otrzymujemy na ekranie:

**Rysunek 4:** Przykładowy stan wyświetlacza

Zapis `"%4u"` oznacza cztery cyfry liczby całkowitej bez znaku.



Funkcja `snprintf()` jest bezpieczniejszą wersją funkcji `sprintf()`, uwzględniającą rozmiar bufora, dzięki czemu nigdy nie nastąpi nadpisanie pamięci za buforem, gdyby wydrukowany napis się w nim nie mieścił.

Wyznaczenie przycisku odpowiadającego pomiarowi

Uzupełnij metodę `Keypad::currentKey()` o wyznaczenie aktualnie wciśniętego przycisku.

Mapowanie między pomiarem ADC a wciśniętym przyciskiem można zrealizować na dwa sposoby:

1. przeliczając każdy pomiar na napięcie, a następnie dopasowując zmierzone napięcie do napięć z klawiatury;
2. wyznaczając już na etapie kompilacji progi w postaci wartości pomiaru ADC i bezpośrednio je porównując.

Drugie rozwiązanie pozwala zaoszczędzić skomplikowanych obliczeń podczas pracy urządzenia i można je zapisać prosto w postaci ciągu instrukcji warunkowych. Korzystając ze wzoru na wartość pomiaru ADC w zależności od mierzonego napięcia V_{IN} i napięcia referencyjnego V_{REF} (tu: napięcia zasilania 5 V):

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}} \quad (1)$$

otrzymujemy:

Wyznacz napięcia progowe dla poszczególnych przycisków z pomocą tabeli w sprawozdaniu.

Wyświetlenie nazwy przycisku

W funkcji `printKey()` należy wypisać na wyświetlaczu w polu `K` nazwę aktualnie wciśniętego przycisku.

Algorithm 1 Rozpoznanie przycisku

```
if measurement <  $0,5/5 \cdot 1024$  then
    return KEY_RIGHT
else if measurement <  $1,5/5 \cdot 1024$  then
    return KEY_UP
else if ... then
    ...
end if
```



```
ADC: 619 K: LEFT
COUNTER:
```

Rysunek 5: Stan wyświetlacza przy wciśniętym przycisku *LEFT* (wartość odczytana z ADC może się nieznacznie różnić)



Możesz użyć ciągu instrukcji **if/else** lub instrukcji **switch/case**.

Zadanie rozszerzone

Celem zadania rozszerzonego jest wykorzystanie klawiatury do zmiany wartości licznika.

Wymagania funkcjonalne

1. W polu `COUNTER` wyświetlana jest wartość licznika `static int16_t counter`.
2. Wciśnięcie przycisku *UP* zwiększa stan licznika o 1.
3. Wciśnięcie przycisku *DOWN* zmniejsza stan licznika o 1.
4. Wciśnięcie przycisku *SELECT* ustawia stan licznika na 0.
5. Wciśnięcie przycisku *LEFT* ustawia stan licznika na wartość minimalną.
6. Wciśnięcie przycisku *RIGHT* ustawia stan licznika na wartość maksymalną.



```
ADC: 619 K: LEFT
COUNTER: -32768
```

Rysunek 6: Przykładowy stan wyświetlacza



Aby ustawić minimalną i maksymalną wartość zmiennej typu `int16_t`, użyj statycznych `INT16_MIN` i `INT16_MAX` z biblioteki standardowej `stdint.h`.

Modyfikacja programu

Reakcja tylko na wciśnięcie przycisku

Uzupełnij metodę `Keypad::oneTimeKey()`, tak by zwracała tylko zmianę stanu przycisku.

Algorithm 2 Reakcja tylko na wciśnięcie przycisku

| | |
|---|-------------------------------------|
| if <i>currentKey()</i> \neq <i>lastKey</i> then | ▷ jeżeli zmienił się stan przycisku |
| <i>lastKey</i> \leftarrow <i>currentKey()</i> | ▷ zapamiętaj nowy stan |
| return <i>lastKey</i> | ▷ zwróć nowy stan |
| else | ▷ jeżeli wciąż ten sam przycisk |
| return KEY_NONE | ▷ zwróć brak przycisku |
| end if | |



Zmienna `lastKey` jest dostępna jako właściwość obiektu `Keypad` (vide plik nagłówkowy `keypad.h`) — nie musisz jej definiować.

Zwiększanie i zmniejszanie licznika

W funkcji `handleCounter()` należy odpowiednio zmieniać i wyświetlać stan licznika.