

Warsztaty AVR

Wyświetlacz LCD

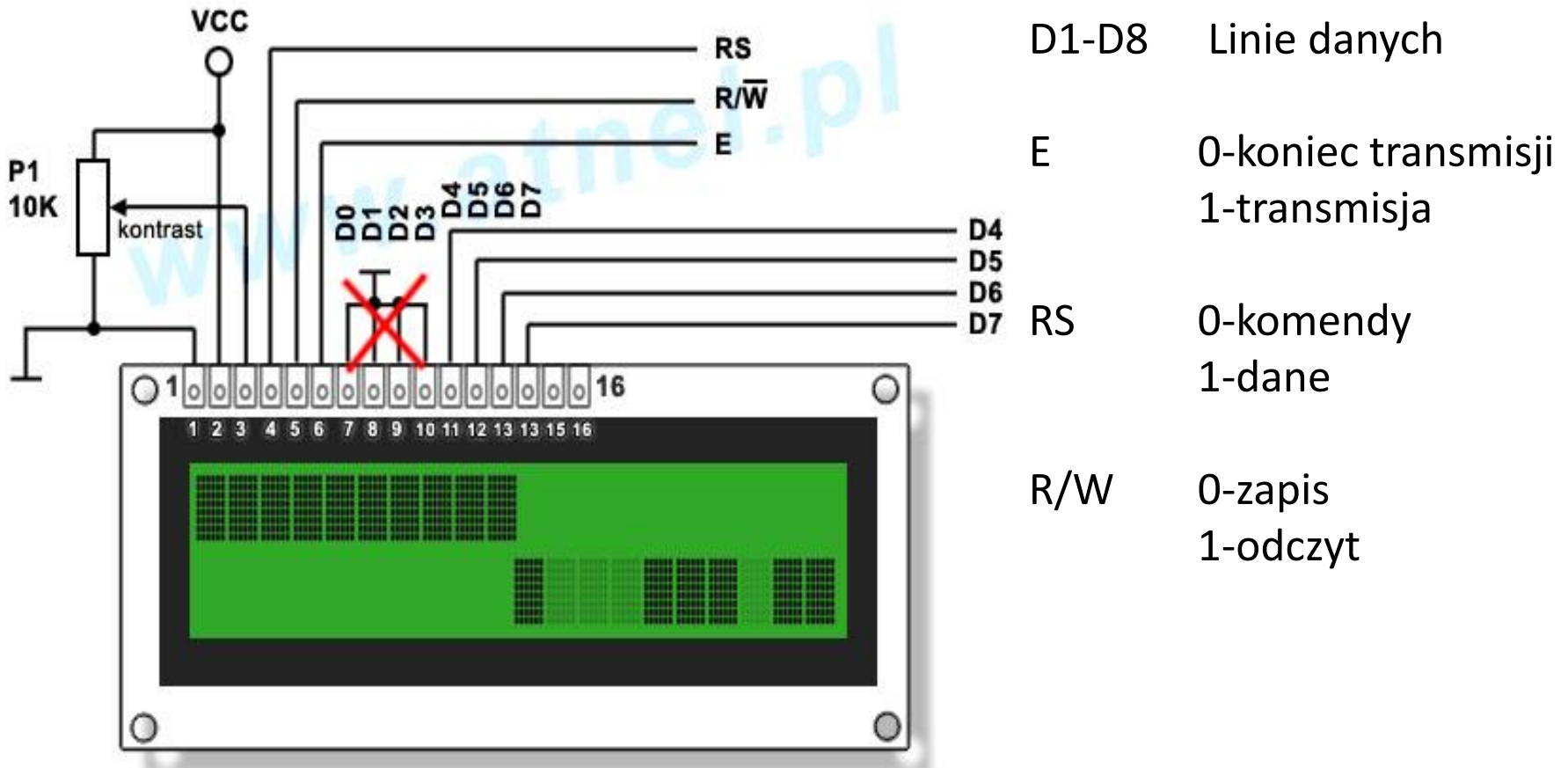
Dariusz Wika



SATLAND
PROTOTYPE



Wyprowadzenia



Komunikacja

- Tryb 4-bitowy oznacza oszczędność linii
- Rezygnując z odczytu danych z wyświetlacza zwieramy pin R/W do masy
- W czasie inicjalizacji informujemy wyświetlacz że korzystamy z trybu 4-bitowego
- Komunikacja z wyświetlaczem odbywa się za pomocą komend dostępnych w dokumentacji
- Rezygnując z odczytu należy pamiętać o stosowaniu programowych opóźnień aby dać czas wyświetlaczowi

Wysłanie danych

Ustawiamy linie E w stan wysoki

Ustawiamy linie danych w odpowiednich stanach

Ustawiamy linie E w stan niski

Ustawiamy linie E w stan wysoki

Ustawiamy linie danych w odpowiednich stanach

Ustawiamy linie E w stan niski

Dajemy opóźnienie żeby wyświetlacz miał czas na reakcję

Makra Preprocesora

Definiujemy w kodzie:

```
#define PORT(x) SPORT(x)
#define SPORT(x) (PORT##x)
```

Potem piszemy:

```
#define LED_PORT D
PORT( LED_PORT ) |= (1<<PD2);
```

Chcemy zmienić stan wyjścia PD2 i piszemy:

```
PORT( LED_PORT ) |= (1<<PD2);
```

Preprocesor Magicznie

zamienia to na:

```
PORTD |= (1<<PD2);
```

Ale czemu nie można napisać?

```
#define PORT(x) (PORT##x)
```

Bo wtedy wyjdzie nam:

```
PORTLED_PORT |= (1<<PD2);
```

Makra Preprocesora

Inline – wstawia kod funkcji w miejsce jej wystąpienia w kodzie

Poniższe makro pozwala wyłączyć kompilację wybranej funkcji:

```
#define CHCEMY_FUNKCJE 1

#if CHCEMY_FUNKCJE == 1
    void funkcja(void)
    {
        głupoty_które_robi funkcja;
    }
#endif
```