

niujących polskie znaki. W naszym przykładzie został on podzielony na mniejsze bloki, w taki sposób, że każdy z nich definiuje jeden znak. Aby załadować blok danych do pamięci posłużymy się instrukcją *MWRITE*. Poniższa tabela przedstawia format instrukcji (Tabela 4.20). Następujące po instrukcji bajty są ładowanymi do pamięci bajtami danych.

Tabela 4.20. Format instrukcji *MWRITE*

	D7	D6	D5	D4	D3	D2	D1	D0
Instrukcja	0	1	0	0	0	0	1	0
P1	pierwszy bajt bloku danych							
P2	drugi bajt bloku danych							
...	...							
...	...							
Pn	n-ty bajt bloku danych							

Inkrementacja adresu dla każdego ładowanego bajtu danych jest automatyczna. Oprócz polskich znaków diaktrycznych, wysłamy również litery *y*, *g*, i *p*. Litery te są ładowane dlatego, ponieważ oryginalne litery mają krótkie „ogonki”, przez co są mało czytelne i źle wyglądają. Przy umieszczeniu kolejnych bajtów, jeden za drugim, nie widać, jak będzie wyglądał definiowany znak. Jeśli umieścimy kolejne bajty jeden pod drugim, wygląd znaku będzie oczywisty. Każda jedyńka oznacza, że odpowiadający jej piksel w matrycy znaku będzie włączony (Przykład 4.29):

Przykład 4.29. Blok danych definiujący literę *y*

```
0b00000000
0b00000000
0b10001000
0b10001000
0b10001000
0b01111000
0b00001000
0b01110000
```

Przykładowa procedura inicjalizacji wyświetlacza nie wykorzystuje wszystkich dostępnych instrukcji. Również zamieszczony opis działania użytych instrukcji i wyświetlacza jest skrócony. Pełny opis, a także przykłady innych konfiguracji można znaleźć w opisie układu SED1335. Niestety, sama procedura inicjalizacji wyświetlacza jest mało przydatna. Potrzebne są jeszcze inne procedury, które umożliwią wyświetlanie tekstów i grafik. Teraz zajmiemy się takimi procedurami. Pisząc procedury obsługi, nie zawsze trzeba na ich początku definiować wielkość kursora, kierunek jego automatycznego przesuwania, definiować adres początku pamięci, gdzie będą ładowane bloki danych itp. Czasami nie jest to potrzebne ze względu na wykonaną już wcześniej procedurę, która określiła dany parametr. W naszych przykładach, w celu zwiększenia uniwersalności procedur oraz w celu uniknięcia trudnych do wykrycia błędów, konfigurację używanych w danej procedurze parametrów będziemy zawsze wykonywać. Rozpocniemy od procedury, która skasuje ekran wyświetlacza. Ekran można kasować nowym obrazem przeznaczonym do wyświetlenia. Najczęściej jednak wyświetlenie nowego obrazu, czy pierwszego obrazu wiąże się jedynie z modyfikacją tylko części ekranu. Wówczas obsługa wyświetlacza jest szybsza, a i bloki danych, które umieszczane są często w pamięci programu, definiujące nowe ekrany, są znacząco mniejsze. Przykład 4.30 przedstawia taką procedurę.