

Spis treści

Rozdział 1

Wyświetlacz matrycowy – pierwszy duży projekt od A do Z	9
1.1. Szczegóły projektu	20
1.2. Pierwsze testy prototypu – na zestawie ATB	22
1.3. Najważniejsza funkcja biblioteki – <code>set_pixel()</code>	35
1.4. Zrozumieć zasady wyświetlania – ekran wirtualny i okno	37
1.5. Optymalizacja kodu przzerwania	40
1.6. Przygotowanie kolejnych podstawowych funkcji graficznych	46
1.7. Fonty – łatwy sposób dzięki programowi Pixel Factory	49
1.8. Pierwsza procedura do wyświetlania fontów	55
1.9. Docelowa platforma sprzętowa – sterownik wyświetlacza	64
1.10. Przeportowanie kodu programu z ATmega32 na ATmega328	66
1.10.1. Zamiana programowej obsługi SPI na sprzętową	69
1.10.2. Przejęcie kontroli nad wyświetlaczem – obsługa sygnału BUSY	72
1.10.3. Wprowadzamy regulację jasności wyświetlacza za pomocą PWM	76
1.11. Programujemy docelowy układ – wielozadaniowy zegar/sterownik WiFi	79
1.11.1. Zaawansowane i wygodne timery programowe	82
1.11.2. Implementacja nowych timerów programowych	88
1.11.3. Obsługa czujnika DS18B20 i timer programowy	94
1.11.4. Podwójne buforowanie i synchronizacja wyświetlacza matrycowego	101
1.11.5. Obsługa RTC, IR, PWM, RS232 – synchronizacja czasu WiFi	108
1.11.6. Nietypowe zastosowania nowych timerów programowych	115
1.11.7. Porządkowanie folderu rozbudowanej biblioteki	117
1.11.8. Warstwy – zaawansowana obsługa wyświetlacza graficznego	119
1.11.9. Składanie warstw w jeden obraz na ekranie	127
1.11.10. Kolejna aktualizacja założeń głównego projektu	130
1.11.11. Realizacja efektów specjalnych – ciekawe animacje	138
1.11.12. Obsługa czujnika ciśnienia BMP180 (biblioteka od zera)	145
1.11.13. Zaawansowane wykorzystanie unii i struktur w C	146
1.11.13.1. Wykorzystanie przesunięć bitowych zamiast operacji dzielenia i mnożenia	154
1.11.13.2. Obliczenia w bibliotece – wykorzystanie wzorów z noty PDF	155
1.11.14. Implementacja biblioteki BMP180 w projekcie zegara	158
1.11.15. Fotorezystor, ADC i automatyczna regulacja jasności wyświetlacza	159
1.11.15.1. Mapowanie zakresów – ciekawe eksperymenty z ADC	165

1.11.16. MSGEQ7 – graficzny analizator widma audio	169
1.11.17. Porządkowanie kodu źródłowego	177
1.11.18. Implementacja obsługi analizatora widma audio w projekcie	180
1.11.19. Przełączanie funkcji zwrotnych do obsługi zdarzeń w kodzie	183
1.11.20. Obsługa różnych wizualizacji analizatora widma – timer programowy	187
1.11.21. Przełączanie widoków wizualizacji za pomocą pilota podczerwieni	192
1.11.22. Komunikacja WiFi (RS232) z komputerem PC lub Androidem	194
1.11.23. Realizacja założeń projektowych komunikacji WiFi/RS232	196
1.11.24. Aplikacja do komunikacji na komputer PC	201
1.11.25. Aplikacja do komunikacji na telefony i tablety z systemem Android (B4A)	202
1.11.26. Podsumowanie – ogólny blokowy schemat projektu	208

Rozdział 2

Zastosowanie pamięci flash jako EEPROM	210
2.1. Podział pamięci flash na sekcje RWW oraz NRWW (BLS)	215
2.2. MkAvrCalculator – nowe narzędzie „flash as EEPROM helper”	220
2.3. Definiowanie adresów dla danych i funkcji – opcje linkera C	222
2.4. Fusebity i wybór adresów – ATmega	225
2.5. Fusebity i wybór adresów – ATtiny	227
2.6. Funkcja zapisu do flash – kod źródłowy (biblioteka)	228
2.7. Praktyczny przykład i test działania biblioteki	233

Rozdział 3

Modem GSM – komunikacja SMS i GPRS	237
3.1. Zasady projektowania PCB do modułów GSM	238
3.2. Moduł ATB-MP-GSM FIBO – zgodny ze złączem ATB-MICROPORT	242
3.2.1. DIGI-LION – moduł przetwornicy z ładowarką akumulatorów Li-Ion oraz Li-Po	244
3.3. Modem Fibocom G510 – pierwsze uruchomienie	245
3.4. Założenia do projektu – testera wszystkich funkcjonalności modemu	250
3.4.1. Podział dostępnych możliwości modemu na kategorie i zadania projektowe	252
3.5. Komunikacja asynchroniczna z modemem za pomocą komend AT	254
3.5.1. Timer programowy – proces realizacji pojedynczej komendy AT	256
3.5.2. Komendy AT – sposób dostępu przez tablicę w pamięci flash	261
3.5.3. Wysyłanie komend AT	267
3.5.4. Oczekiwanie na rezultat komend AT w sposób nieblokujący	270
3.5.5. Parsowanie rezultatów komend AT	274
3.6. Pierwsze zastosowanie komend AT w programie	275
3.7. Pierwsze testy i debugowanie projektu w zestawie ATB	286
3.8. Logowanie modemu G510 do operatora GSM	291
3.8.1. Wprowadzanie kodu PIN do karty SIM – AT+CPIN	295
3.9. Garść informacji ogólnych na temat wiadomości SMS	299
3.10. Pierwszy SMS – w trybie tekstowym	301

3.11.	Wstęp do odbioru wiadomości SMS	308
3.11.1.	Najważniejsza komenda AT związana z obsługą wiadomości SMS	308
3.11.2.	Obserwacja przychodzących wiadomości w terminalu	309
3.11.3.	Analiza odebranych danych w terminalu – SMS przychodzący	310
3.11.4.	Odbiór wiadomości SMS typu tekstowego	313
3.12.	Jak nie należy tworzyć projektu – przykład złego kodu	316
3.13.	Porządkowanie kodu, podział na część biblioteczną i główny kod projektu	318
3.13.1.	Modyfikacja biblioteki MK_USART – dodanie ukrytego callbacka	321
3.13.2.	Rozbudowa biblioteki GSM – parsowanie danych z modemu	324
3.13.3.	Separacja funkcji wysyłania komend AT w bibliotece i poza biblioteką	337
3.13.4.	Stos w mikrokontrolerze a obsługa oparta na zdarzeniach i timerach programowych	340
3.14.	Wiadomości SMS w trybie PDU	341
3.14.1.	Co to jest PDU?	341
3.14.2.	Powody, dla których warto korzystać z PDU	343
3.14.3.	Skład ramki/pakietu PDU	343
3.14.4.	Numer telefonu w PDU	345
3.14.5.	Rozpoczynamy dekodowanie TPDU	346
3.14.6.	TPDU i pierwszy oktet „PDU type” – przejmujemy władzę nad wiadomością SMS	347
3.14.7.	TPDU – pozostałe pola pakietu	349
3.14.8.	Wiadomość nadawana	349
3.14.9.	Wiadomość odebrana	351
3.14.10.	Ostatnie dwa wspólne pola TPDU – UDL i UD	351
3.14.11.	Sklejanie SMS-ów w jedną dużą wiadomość – UDH	352
3.14.12.	Maksymalne liczby znaków w wiadomościach SMS	358
3.14.13.	Zasada kodowania znaków do formatu 7-bit (septety)	358
3.15.	Nadawanie wiadomości SMS w PDU – nowa funkcja	362
3.16.	Odbiór wiadomości SMS w trybie PDU	377
3.17.	Raporty doręczeń wiadomości SMS – obsługa w trybie PDU	386
3.18.	ECHO – dlaczego zawsze warto je wyłączać?	391
3.19.	Komunikacja GPRS. Wprowadzenie	394
3.19.1.	Nawiązanie połączenia GPRS	395
3.19.2.	SMTP – wysyłanie wiadomości e-mail	406
3.19.3.	FTP – operacje na plikach serwera	417
3.19.4.	HTTP i komunikacja poprzez skrypty PHP	429
3.19.5.	TCP i UDP	440
3.19.5.1.	Komendy AT do obsługi TCP i UDP	441
3.19.5.2.	Zasady wysyłania danych przez TCP i UDP	444
3.19.5.3.	Zasady odbierania danych TCP i UDP	446
3.19.5.4.	Nowy moduł biblioteki do obsługi UDP	448
3.19.5.5.	Przygotowanie serwera UDP do testów na komputer PC	457
3.19.5.6.	Program testowy – serwer UDP	460
3.19.5.7.	Moduł ATNEL-WIFI232-T jako serwer UDP	462

3.20. Połączenia audio – funkcjonalność telefonu komórkowego	463
3.20.1. Obsługa audio w bibliotece MK_GSM	465
3.20.1.1. Modyfikacja funkcji do wysyłania wiadomości SMS	
– mk_sms.c	474
3.20.1.2. Obsługa połączeń głosowych z poziomu programu	476
3.20.2. Obsługa torów audio modemu za pomocą komend AT	480
3.21. Funkcje przycisków pilota IR w projektach	483
3.22. Dalsze możliwości rozwoju projektu	485
3.23. Podsumowanie konfiguracji – kosmetyka inicjalizacji	488