

Sommaire

1. Organisation du livre7	3.4 Compilation 27
<hr/>	3.5 Utilisation d'un langage plus « parlant » 27
2. Présentation des PIC	3.6 Correspondance entre Pascal et assembleur 28
<hr/>	4. Présentation de mikroPascal
2.1 Au début était... la logique câblée ..9	4.1 Constitution d'un programme ... 31
2.2 Arrivée du microprocesseur9	4.1.1 Structure d'un programme 32
2.3 Arrivée du microcontrôleur10	4.1.2 Les variables 34
2.4 Différentes familles de PIC11	4.1.3 Quelques règles simples 35
2.4.1 PIC « de base » 8 bits (PIC10, PIC12, PIC16, PIC18)12	4.1.4 Les opérateurs 36
2.4.2 PIC « évolués » 16 bits (PIC24, dsPIC30, dsPIC33)15	4.1.5 Les fonctions spéciales 38
2.4.3 PIC « haut de gamme » 32 bits (PIC32)18	4.2 Introduction à mikroPascal 39
2.4.4 Différentes versions d'un « même » PIC19	4.3 Principes de base 41
2.5 Les mots qui fâchent20	4.4 Principes avancés 43
2.5.1 Bit, octet, mot, entier... (bit, <i>byte</i> , <i>word</i> , <i>integer</i> ...)21	4.4.1 Simulation 43
2.5.2 Registres (<i>registers</i>)21	4.4.2 Débogage ICD 45
2.5.3 Entrées/sorties (<i>inputs/outputs</i>)23	4.5 mikroPascal et programmeurs .. 45
2.5.4 Mémoire (<i>memory</i>)23	<hr/>
2.5.5 Bus24	5. Configuration d'un PIC
2.5.6 I2C / SPI / OneWire24	<hr/>
3. Assembleur et langage machine	5.1 Mot de configuration (<i>Config Word</i>) 48
<hr/>	5.2 FOOSC (<i>Oscillator</i>) – horloge/oscillateur 51
3.1 Exécution de commandes25	5.2.1 Vitesse d'exécution des instructions 52
3.2 Langage machine26	
3.3 Langage assembleur26	

5.2.2	Localisation des broches affectées à horloge	52
5.2.3	Différents modes d'horloge	53
5.2.4	PLL et multiplication de fréquence ..	64
5.2.5	Blocage de l'oscillateur (fonction SLEEP / Power Down)	64
5.2.6	Timers et compteurs	67
5.2.7	Fréquence et tension d'alimentation ..	71
5.2.8	Overclocking	71
5.2.9	Remarques diverses	72
5.3	MCLR (Master Clear Reset) – Reset général	72
5.3.1	Reset manuel	73
5.3.2	Reset automatique	76
5.4	WDT (WatchDog Timer) – Chien de garde	76
5.5	BOR (Brown-Out Reset) – Détection de chute de tension	80
5.6	POR (Power-On Reset) – Reset à la mise sous tension	85
5.7	PWRT (Power-up Timer) – Temporisateur de démarrage	86
5.8	LVP (Low Voltage Programming) – Programmation en basse tension ..	87
5.9	CP (Code Protection) – Protections de la mémoire	89
5.10	Configuration des broches d'entrée/sortie	91
5.10.1	Des broches à tout faire	91
5.10.2	L'heure du choix... ..	94

6. Programmation d'un PIC

6.1	Cinq fils suffisent pour programmer un PIC	10
6.2	Modes de programmation HVP ou LVP	10
6.3	VPP-First ou VDD-First ?	10
6.4	Programmation sur support (off-board)	10
6.4.1	Programmateurs simples	10
6.4.2	Programmateurs évolués (platines de développement)	10
6.5	Programmation sur circuit final (ICSP)	11
6.5.1	Câblage d'un connecteur ICSP	114
6.6	Programmation avec bootloader ..	118
6.6.1	Utilisation d'un bootloader COM/RS232	121
6.6.2	Utilisation d'un bootloader USB ..	123
6.6.3	Autres types de bootloader	124
6.6.4	Plus besoin de programmeur ? ...	124
6.7	Alors, ICSP ou bootloader ?	125
6.8	Choix du programmeur	125
6.8.1	Programmeur à faire soi-même ..	126
6.8.2	Programmateurs universels ?	132
6.8.3	Durée de programmation d'un PIC ..	133

7. Interruptions

7.1	Interruption de Timer	142
7.2	Interruption via les ports d'entrée	150
7.3	Autres types d'interruption	156

8.	Utilisation des entrées/sorties	8.7	Mesure d'une fréquence 222
		8.7.1	Mesure d'une fréquence après conversion N/A simple 222
		8.7.2	Mesure directe d'une fréquence . . . 225
8.1	Activation d'une sortie162	8.8	Production d'une tension analogique 227
8.1.1	Des sorties... mais aussi des entrées 162	8.8.1	Production d'une tension analogique avec un réseau R-2R 228
8.1.2	Clignotant à LED simple167	8.8.2	Production d'une tension analogique avec un DAC intégré 230
8.1.3	Gradateur de lumière à LED168	8.8.3	Production d'une tension analogique avec un signal PWM 232
8.1.4	Pilotage d'un relais ou d'un moteur 172	8.8.4	Production d'une tension analogique par moyen détourné 236
8.2	Production d'un signal sonore . . .174	8.9	Usage de capteurs et détecteurs . . 237
8.2.1	Signal sonore continu175	8.9.1	Capteur de contact en tout ou rien . 237
8.2.2	Signal sonore bref (bip)176	8.9.2	Capteur de lumière en tout ou rien . 240
8.2.3	Signal modulé (sirène)178	8.9.3	Capteur de lumière de type proportionnel 241
8.2.4	Mélodie (suite de notes)179	8.9.4	Capteur de température analogique 242
8.2.5	Boîte à rythmes180	8.9.5	Capteur de température numérique 251
8.2.6	Lecture de fichiers WAV/RAW . . .186	8.9.6	Capteur d'humidité analogique . . . 253
8.2.7	Lecture de fichiers MP3 (approche) 188	8.9.7	Capteur d'humidité numérique . . . 257
8.3	Capture de signaux logiques190	8.9.8	Capteur de pression 257
8.3.1	Capture de signaux lents191	8.9.9	Capteur capacitif et touche sensitive 261
8.3.2	Capture de signaux rapides197	8.9.10	Capteur de position (accéléromètre) à 3 axes 270
8.3.3	Encodeur optique202	8.10	Multiplexages des entrées/sorties 272
8.4	Utilisation des comparateurs206	8.10.1	Multiplexage de sorties 273
8.4.1	Comparaison de deux tensions externes210	8.10.2	Multiplexage d'un clavier et d'un afficheur à LED 279
8.4.2	Comparaison d'une tension externe avec la tension de référence interne210	8.11	Affichage sur écran LCD 283
8.5	Mesure d'une tension212	8.11.1	Principe général de fonctionnement 283
8.5.1	Mesure de tension(s) avec CAN interne213	8.11.2	Affichage LCD de type texte 284
8.5.2	Mesure de tension(s) sans CAN interne218	8.11.3	Affichage LCD de type graphique . 287
8.6	Mesure d'un rapport cyclique . . .218		
8.6.1	Mesure du rapport cyclique après intégration (analogique)218		
8.6.2	Mesure directe du rapport cyclique (numérique)220		

9. Gestion des données

9.1	Lecture/écriture en EEPROM . . .293	9.6.5	Lecture/écriture avec mémoire EEPROM externe (I2C) 358
9.1.1	Quel type de données peut-on sauver en EEPROM ?294	9.6.6	Convertisseur A/N à 8 voies avec MCP3208 (SPI) 361
9.1.2	Écriture/lecture d'une donnée de 8 bits (octet/byte)294	9.6.7	Lecture/écriture de valeurs logiques avec MCP23S08 (SPI) 364
9.1.3	Écriture/lecture d'une donnée de 16 bits (mot/word)297	9.6.8	Utilisation simultanée de composants I2C et SPI 368
9.1.4	Écriture/lecture d'une donnée de 32 bits (mot long/longword) . . .301	9.7	Liaison USB 368
9.1.5	Écriture d'une donnée de 7 bits ou moins303	9.7.1	Concepts de base 370
9.2	Données RS232305	9.7.2	PIC 18F2550 et liaison USB fonctionnelle 378
9.2.1	Émission et réception de données RS232307	9.7.3	PIC 18F2550 et utilisation pratique 389
9.2.2	Utilisation d'un <i>buffer</i> circulaire . .310	9.7.4	PIC 18F4550 et utilisation pratique 395
9.2.3	Différences entre RS232 et RS485 .314	9.8	Liaison Ethernet 401
9.2.4	Interface UART/USB316	9.8.1	Concept de base 401
9.3	Données MIDI317	9.8.2	18F45K22, ENC28J60 et liaison Ethernet fonctionnelle 403
9.3.1	Émission de données MIDI318	9.9	Liaison OneWire (1-Wire) 413
9.3.2	Réception de données MIDI321	9.9.1	Mode d'alimentation parasite (<i>parasitic power</i>) 413
9.4	Données RC5/RC6324	9.9.2	Protocole de communication 416
9.4.1	Émission de données RC5324	Annexe 1 – Binaire, décimal et hexadécimal 419	
9.4.2	Réception de données RC5327	Annexe 2 – Convertisseurs analogiques/numériques 435	
9.5	Données DMX331	Annexe 3 – Choix du « bon » microcontrôleur 439	
9.5.1	Émission de données DMX334	Annexe 4 – Logiciels 441	
9.5.2	Réception de données DMX336	Annexe 5 – Ressources 443	
9.6	Liaisons SPI et I2C340		
9.6.1	Présentation du bus I2C340		
9.6.2	Présentation du bus SPI347		
9.6.3	Lecture/écriture de valeurs logiques avec PCF8574 (I2C)350		
9.6.4	Lecture/écriture de valeurs analogiques avec PCF8591 (I2C)353		