



Découvrez notre Chaîne YouTube "[Ingénierie et Projets](#)"

Découvrez notre Chaîne Secondaire "[Information Neuronale et l'Ingénierie du Cerveau](#)"

Titre: La programmation des PIC la programmation des pic avec le PIC16F84

Auteurs: BIGONOFF

Ecole: Néant

Résumé: Et voilà, nous sommes partis ensemble pour cette grande aventure qu'est la programmation des PIC®. Je vais tenter de rester le plus concret possible, mais, cependant, une certaine part de théorie est indispensable pour arriver au but recherché. Je vais donc commencer ce petit [cours](#) par un rappel sur les systèmes de numération. Ca y est, j'en vois qui râlent déjà. Mais je suis sûr que vous comprendrez qu'il est impossible de programmer sérieusement un [microcontrôleur](#) sans savoir ce qu'est un bit, ou comment convertir les notations décimales en hexadécimales. Rassurez-vous, je vais faire bref, et nous pourrons très rapidement aborder le sujet qui nous intéresse tant. Si vous êtes déjà un « pro », vous pouvez sauter le premier chapitre et passer directement au suivant.

N'hésitez jamais à me faire part de vos remarques, ni à me signaler les erreurs qui m'auraient échappées (www.bigonoff.org). Répercutez les infos que vous trouverez ici, traduisez le document dans une autre langue ou un autre format. Simplement, dans ce cas, veuillez respecter les désirs de l'auteur en fin d'ouvrage et faites moi parvenir un exemplaire de votre travail. Ceci pour permettre d'en faire profiter le plus grand nombre (bigocours@hotmail.com).

Extrait du sommaire:

1. INTRODUCTION. 9
2. LES SYSTEMES DE NUMERATION . 11
- 2.1 LE SYSTEME DECIMAL 11



- 2.2 LE SYSTEME BINAIRE . 11
- 2.3 LE SYSTEME HEXADECIMAL. 13
- 2.4 LES OPERATIONS 14
- 2.5 LES NOMBRES SIGNES 15
- 2.6 LES OPERATIONS BOOLEENNES. . 16
 - 2.6.1 Le complément . 16
 - 2.6.2 La fonction « ET » ou « AND » 16
 - 2.6.3 La fonction « OU » ou « OR » . 17
 - 2.6.4 La fonction « OU EXCLUSIF » ou « Exclusif OR » ou « XOR » 18
- 2.7 UN MOT SUR LES UNITES 18
- 3. COMPOSITION ET FONCTIONNEMENT DES PIC® 21
 - 3.1 QU'EST-CE QU'UN PIC® ? . 21
 - 3.2 LES DIFFERENTES FAMILLES DES PIC® 22
 - 3.3 IDENTIFICATION D'UN PIC® 23
 - 3.4 ORGANISATION DU 16F84 24
 - 3.4.1 La mémoire programme. 24
 - 3.4.2 La mémoire [Eeprom](#). 24
 - 3.4.3 La mémoire Ram 25
- 4. ORGANISATION DES INSTRUCTIONS. 27
 - 4.1 GENERALITES. 27
 - 4.2 LES TYPES D'INSTRUCTIONS. 27
 - 4.2.1 Les instructions « orientées octet » 27
 - 4.2.2 Les instructions « orientées bits » 28
 - 4.2.3 Les instructions générales 28
 - 4.2.4 Les sauts et appels de sous-routines. 28
 - 4.3 PANORAMIQUE DES INSTRUCTIONS 28
 - 4.4 LES INDICATEURS D'ETAT 30
 - 4.4.1 L'indicateur d'état « Z » 31
 - 4.4.2 L'indicateur d'état « C » 31
- 5. LES DEBUTS AVEC MPLAB® 33
 - 5.1 PREPARATION A L'UTILISATION . 33



- 5.2 CREATION DE NOTRE PREMIER PROJET . 34
- 6. ORGANISATION D'UN FICHIER « .ASM » . 41
 - 6.1 LES COMMENTAIRES 41
 - 6.2 LES DIRECTIVES . 41
 - 6.3 LES FICHIERS « INCLUDE » . 42
 - 6.4 LA DIRECTIVE _CONFIG. 42
 - 6.5 LES ASSIGNATIONS. 43
 - 6.6 LES DEFINITIONS 44
 - 6.7 LES MACROS 44
 - 6.8 LA ZONE DES VARIABLES . 45
 - 6.9 LES ETIQUETTES. 45
 - 6.10 LA DIRECTIVE « ORG » . 46
 - 6.11 LA DIRECTIVE « END » ET LA FIN D'UN PROGRAMME 46
- 7. REALISATION D'UN PROGRAMME . 49
 - 7.1 CREATION DE NOTRE PREMIER PROGRAMME 49
 - 7.2 L'ASSEMBLAGE D'UN PROGRAMME 50
- 8. LA SIMULATION D'UN PROGRAMME. 53
 - 8.1 LANCEMENT ET PARAMETRAGE DU SIMULATEUR. 53
 - 8.2 EXPLICATION DES REGISTRES FONDAMENTAUX . 55
 - 8.2.1 Les registres « PCL » et « PCLATH » . 55
 - 8.2.2 Le registre « W » 56
 - 8.2.3 Le registre « STATUS » 56
 - 8.3 LANCEMENT DE LA SIMULATION 57
- 9. LE JEU D'INSTRUCTIONS . 63
 - 9.1 L'INSTRUCTION « GOTO » (ALLER À) . 63
 - 9.2 L'INSTRUCTION « INCF » (INCREMENT FILE). 64
 - 9.3 L'INSTRUCTION « DECF » (DECREMENT FILE) 65
 - 9.4 L'INSTRUCTION « MOVLW » (MOVE LITERAL TO W). 65
 - 9.5 L'INSTRUCTION « MOVF » (MOVE FILE) . 66
 - 9.6 L'INSTRUCTION « MOVWF » (MOVE W TO FILE) 67
 - 9.7 L'INSTRUCTION « ADDLW » (ADD LITERAL AND W) 67



- 9.8 L'INSTRUCTION « ADDWF » (ADDW AND F) . 68
- 9.9 L'INSTRUCTION « SUBLW » (SUBTRACT W FROM LITERAL). 68
- 9.10 L'INSTRUCTION « SUBWF » (SUBTRACT W FROM F) 71
- 9.11 L'INSTRUCTION « ANDLW » (AND LITERAL WITH W) 71
- 9.12 L'INSTRUCTION « ANDWF » (ANDW WITH F) 72
- 9.13 L'INSTRUCTION « IORLW » (INCLUSIVE OR LITERAL WITH W) . 73
- 9.14 L'INSTRUCTION « IORWF » (INCLUSIVE ORW WITH FILE) 73
- 9.15 L'INSTRUCTION « XORLW » (EXCLUSIVE OR LITERAL WITH W) 74
- 9.16 L'INSTRUCTION « XORWF » (EXCLUSIVE ORW WITH F) 74
- 9.17 L'INSTRUCTION « BSF » (BIT SET F) . 75
- 9.18 L'INSTRUCTION « BCF » (BIT CLEAR F) 75
- 9.19 L'INSTRUCTION « RLF » (ROTATE LEFT THROUGH CARRY) . 76
- 9.20 L'INSTRUCTION « RRF » (ROTATE RIGHT THROUGH CARRY). 77
- 9.21 L'INSTRUCTION « BTFSC » (BIT TEST F, SKIP IF CLEAR) 78
- 9.22 L'INSTRUCTION « BTFSS » (BIT TEST F, SKIP IF SET). 80
- 9.23 L'INSTRUCTION « DECFSZ » (DECREMENT F, SKIP IF Z). 80
- 9.24 L'INSTRUCTION « INCFNZ » (INCREMENT F, SKIP IF ZERO). 82
- 9.25 L'INSTRUCTION « SWAPF » (SWAP NIBBLES IN F) 82
- 9.26 L'INSTRUCTION « CALL » (CALL SUBROUTINE) 83
- 9.27 L'INSTRUCTION « RETURN » (RETURN FROM SUBROUTINE) . 84
- 9.28 L'INSTRUCTION « RETLW » (RETURN WITH LITERAL IN W) . 86
- 9.29 L'INSTRUCTION « RETFIE » (RETURN FROM INTERRUPT). 87
- 9.30 L'INSTRUCTION « CLRF » (CLEAR F). 87
- 9.31 L'INSTRUCTION « CLRW » (CLEAR W) 88
- 9.32 L'INSTRUCTION « CLRWDN » (CLEARWATCHDOG) 88
- 9.33 L'INSTRUCTION « COMF » (COMPLEMENT F) 88
- 9.34 L'INSTRUCTION « SLEEP » (MISE EN SOMMEIL) . 89
- 9. 35 L'INSTRUCTION « NOP » (NO OPERATION) 89
- 9.36 LES INSTRUCTIONS OBSOLETES 90
- 10. LES MODES D'ADRESSAGE 91
- 10.1 L'ADRESSAGE LITTERAL OU IMMEDIAT 91



- 10.2 L'ADRESSAGE DIRECT 91
- 10.3 L'ADRESSAGE INDIRECT 92
 - 10.3.1 Les registres FSR et INDF . 92
- 10.4 QUELQUES EXEMPLES 93
- 11. REALISATION D'UN PROGRAMME EMBARQUE. 95
 - 11.1 LE MATERIEL NECESSAIRE . 95
 - 11.2 MONTAGE DE LA PLATINE D'ESSAIS . 96
 - 11.3 CREATION DU PROJET. 97
 - 11.4 EDITION DU FICHER SOURCE . 97
 - 11.5 CHOIX DE LA CONFIGURATION . 97
 - 11.6 LE REGISTRE OPTION. 99
 - 11.7 EDITION DU PROGRAMME. 100
 - 11.8 LE REGISTRE PORTA. 103
 - 11.8.1 Fonctionnement particulier des PORTS. 105
 - 11.9 LE REGISTRE TRISA 106
 - 11.10 LES REGISTRES PORTB ET TRISB. 107
 - 11.11 EXEMPLE D'APPLICATION 108
 - 11.12 LA ROUTINE D'INITIALISATION. 108
 - 11.13 LES RESULTATS DE L'ASSEMBLAGE 112
 - 11.14 LE PROGRAMME PRINCIPAL 112
 - 11.15 LA SOUS-ROUTINE DE TEMPORISATION 113
- 12. LES **INTERRUPTIONS**. 119
 - 12.1 QU'EST-CE QU'UNE INTERRUPTION ?. 119
 - 12.2 MECANISME GENERAL D'UNE INTERRUPTION 120
 - 12.3 MECANISME D'INTERRUPTION SUR LES PIC® 121
 - 12.4 LES SOURCES D'INTERRUPTIONS DU 16F84 123
 - 12.5 LES DISPOSITIFS MIS EN OEUVRE 123
 - 12.6 LE REGISTRE INTCON (INTERRUPT CONTROL). 125
 - 12.7 SAUVEGARDE ET RESTAURATION DE L'ENVIRONNEMENT 127
 - 12.7.1 Les registres à sauvegarder . 128
 - 12.7.2 La méthode de sauvegarde. 128



- 12.7.3 La méthode de restauration 129
- 12.7.4 OPERATIONS SUR LE REGISTRE STATUS 131
- 12.7.5 Particularité de l'instruction « RETFIE ». 132
- 12.8 UTILISATION D'UNE ROUTINE D'INTERRUPTION. 133
- 12.9 ANALYSE DE LA ROUTINE D'INTERRUPTION . 136
- 12.10 ADAPTATION DE LA ROUTINE D'INTERRUPTION . 138
- 12.11 L'INITIALISATION. 139
- 12.12 CONSTRUCTION DU PROGRAMME PRINCIPAL 140
- 12.13 CONSTRUCTION DE LA ROUTINE D'INTERRUPTION . 141
- 12.14 PASSAGE AU SIMULATEUR D'UNE ROUTINE D'INTERRUPTION 142
- 12.15 PREMIERE CORRECTION : [RESET](#) DU FLAG 146
- 12.16 SE METTRE A L'ECHELLE DE TEMPS DU PIC® 147
- 12.17 LE PROBLEME DE L'ANTI-REBOND 147
- 12.18 FINALISATION DU PROGRAMME 148
- 12.19 REMARQUES IMPORTANTES 151
- 12.20 CONCLUSIONS 152
- 13. LE TIMER 0 155
- 13.1 LES DIFFERENTS MODES DE FONCTIONNEMENT 155
- 13.2 LE REGISTRE TMR0. 155
- 13.3 LES METHODES D'UTILISATION DU TIMER0 . 155
- 13.3.1 Le mode de lecture simple 156
- 13.3.2 Le mode de scrutation du flag 156
- 13.3.3 Le mode d'interruption. 157
- 13.3.4 Les méthodes combinées 157
- 13.4 LE PREDIVISEUR. 157
- 13.5 APPLICATION PRATIQUE DU TIMER0. 160
- 13.5.1 Préparations. 160
- 13.5.2 L'initialisation 161
- 13.5.3 La routine d'interruption . 162
- 13.6 MODIFICATION DES REGISTRES DANS LE SIMULATEUR. 163
- 13.7 MISE EN PLACE SUR LA PLATINE D'ESSAIS 164



- 13.8 PREMIERE AMELIORATION DE LA PRECISION 164
- 13.9 SECONDE AMELIORATION DE LA PRECISION . 165
- 13.10 LA METHODE DITE « DE RATTRAPAGE ». 165
- 13.11 LA METHODE HARDWARE - ADAPTATION DE L'HORLOGE 167
- 13.12 LA METHODE DE LUXE : LA DOUBLE [HORLOGE](#) 167
- 13.13 EXEMPLE D'UTILISATION DE 2 INTERRUPTIONS 168
- 13.13 CONCLUSION 169
- 14. LES ACCES EN MEMOIRE « EEPROM » 171
- 14.1 TAILLE ET LOCALISATION DE LA MEMOIRE « EEPROM ». 171
- 14.2 PREPARATION DU PROGRAMME 171
- 14.3 INITIALISATION DE LA ZONE EEPROM. 173
- 14.4 LE REGISTRE EEDATA 175
- 14.5 LE REGISTRE EEADR 175
- 14.6 LE REGISTRE EECON1 175
- 14.7 LE REGISTRE EECON2 176
- 14.8 ACCES EN LECTURE DANS LA MEMOIRE « EEPROM » 176
- 14.9 L'ACCES EN ECRITURE A LA ZONE EEPROM 177
- 14.10 UTILISATION PRATIQUE DE LA MEMOIRE « EEPROM » 179
- 14.11 SECURISATION DES ACCES EN MEMOIRE « EEPROM » . 182
- 14.12 CONCLUSION 183
- 15. LE WATCHDOG 185
- 15.1 LE PRINCIPE DE FONCTIONNEMENT 185
- 15.2 LE PREDIVISEUR ET LE WATCHDOG 186
- 15.3 LES ROLES DU WATCHDOG. 186
- 15.4 UTILISATION CORRECTE DU WATCHDOG 187
- 15.5 CE QU'IL NE FAUT PAS FAIRE 188
- 15.6 MESURE DU TEMPS REEL DU WATCHDOG . 188
- 15.7 SIMULATION DU PLANTAGE D'UN PROGRAMME . 190
- 15.7.1 Correction avec utilisation du watchdog . 191
- 15.8 CHOIX DE LA VALEUR DU PREDIVISEUR 192
- 15.9 TEMPS TYPIQUE, MINIMAL, ET MAXIMUM. 192



15.10 CONCLUSION	192
16. LE MODE SLEEP	195
16.1 PRINCIPE DE FONCTIONNEMENT .	195
16.2 LA SORTIE DU MODE « SLEEP »	195
16.3 REVEIL AVEC GIE HORS SERVICE.	196
16.4 REVEIL AVEC GIE EN SERVICE .	196
16.5 MISE EN SOMMEIL IMPOSSIBLE.	196
16.6 UTILISATION DU MODE « SLEEP ».	197
REMARQUE	198
16.7 CAS TYPIQUES D'UTILISATION	198
16.7 POUR UNE CONSOMMATION MINIMALE	198
16.8 CONCLUSION	199
17. LE RESTE DU DATASHEET.	201
17.1 LA STRUCTURE INTERNE	201
17.2 LA SEQUENCE DE DECODAGE .	201
17.3 ORGANISATION DE LA MEMOIRE	201
17.4 LES REGISTRES SPECIAUX.	202
17.5 L'ELECTRONIQUE DES PORTS .	202
17.6 LE REGISTRE DE CONFIGURATION	202
17.7 LES DIFFERENTS TYPES D'OSCILLATEURS.	203
17.7.1 La précision de l'oscillateur	204
17.8 LE RESET	205
17.9 LA MISE SOUS TENSION	206
17.10 CARACTERISTIQUES ELECTRIQUES .	207
17.11 PORTABILITE DES PROGRAMMES	207
17.12 LES MISES A JOUR DES COMPOSANTS	208
17.13 CONCLUSION	209
18. ASTUCES DE PROGRAMMATION.	211
18.1 LES COMPARAISONS.	211
18.2 SOUSTRAIRE UNE VALEUR DE W.	212
18.3 LES MULTIPLICATIONS .	212



- 18.4 MULTIPLICATION PAR UNE CONSTANTE . 215
- 18.5 ADRESSAGE INDIRECT POINTANT SUR 2 ZONES DIFFERENTES. 216
- 18.6 LES TABLEAUX EN MEMOIRE PROGRAMME. 217
- 18.7 LES VARIABLES LOCALES. 221
 - 18.7.1 Détermination des variables locales 222
 - 18.7.2 Construction sans variables locales 222
 - 18.7.3 Construction avec variables locales. 222
- 18.8 DIVISION PAR UNE CONSTANTE 223
- 18.9 REMPLISSAGE D'UNE ZONE MEMOIRE 223
- 18.10 CONCLUSION 224
- 19. UTILISATION DE ROUTINES DANS UN FICHIER SEPRE. 225
 - 19.1 QUESTIONS ET POINT DE DEPART . 225
 - 19.2 UTILISATION DIRECTE DES ROUTINES DANS LE FICHIER . 225
 - 19.3 ENCAPSULATION DANS DES MACROS SIMPLES 228
 - 19.4 MÉTHODE FINALISÉE 230
 - 19.5 CONCLUSION 231
- 20. LA NORME ISO 7816 233
 - 20.1 SPECIFICITES UTILES DE LA NORME ISO 7816 233
 - 20.1.1 Les commandes ISO 7816 234
 - 20.1.2 Le protocole d'échange d'informations . 235
 - 20.2 LES LIAISONS SERIE ASYNCHRONES . 236
 - 20.2.1 Le start-bit 236
 - 20.2.2 Les bits de données. 236
 - 20.2.3 Le bit de parité 237
 - 20.2.4 Le stop-bit 237
 - 20.2.5 Vitesse et débit 237
 - 20.3 ACQUISITION DES BITS . 238
 - 20.4 CARACTERISTIQUE DES CARTES « STANDARD » . 239
 - 20.5 CREATION ET INITIALISATION DU PROJET. 239
 - 20.6 LA BASE DE TEMPS . 240
 - 20.7 RECEPTION D'UN OCTET. 241



20.8 L'EMISSION D'UN CARACTERE . 243

20.9 INITIALISATION 245

20.10 ENVOI DE L'ATR . 246

20.11 L'ENVOI DU STATUT. 247

20.12 RECEPTION DE LA CLASSE 248

20.13 RECEPTION DE INS, P1, P2, ET LEN 248

20.14 CONTROLE DE L'INSTRUCTION REÇUE 249

20.15 TRAITEMENT D'UNE INSTRUCTION . 249

20.16 LES VARIABLES 250

20.17 CONCLUSION 251

ANNEXE1 : QUESTIONS FREQUEMMENT POSEES (F.A.Q.). 253

A1.1 JE TROUVE QUE 8 SOUS-PROGRAMMES, C'EST PEU 253

A1.2 JE N'UTILISE QUE 8 IMBRICATIONS, ET POURTANT MON PROGRAMME PLANTE. . 253

A1.3 MON PROGRAMME SEMBLE NE JAMAIS SORTIR DES INTERRUPTIONS . 253

A1.4 JE N'ARRIVE PAS A UTILISER LE SIMULATEUR, LES OPTIONS N'APPARAISSENT PAS 253

A1.5 JE REÇOIS UN MESSAGE D'ERREUR EOF AVANT INSTRUCTION END . 254

A1.6 COMMENT DESASSEMBLER UN FICHER « .HEX » ? 254

A1.7 UTILISATION DES MINUSCULES ET DES MAJUSCULES. 254

A1.8 LE CHOIX D'UN PROGRAMMATEUR 254

A1.9 J'AI UNE ERREUR DE « STACK ». 255

A1.10 QUELLES SONT LES DIFFERENCES ENTRE 16F84 ET 16F84A ?. 256

A1.11 J'AI UNE ERREUR 173 LORS DE L'ASSEMBLAGE . 257

A1.12 LE PIC16F84 EST OBSOLETE, POURQUOI NE PAS UTILISER LE 16F628 ? 257

A1.13 J'UTILISE UNE VERSION DE MPLAB® PLUS RECENTE. 258

A1.14 MON PIC VIERGE N'OSCILLE PAS . 258

CONTRIBUTION SUR BASE VOLONTAIRE 261

UTILISATION DU PRESENT DOCUMENT . 262

[Cours Microcontrôleur microprocesseur 65](#)

Télécharger le fichier PDF: [La programmation des PIC la programmation des pic avec le PIC16F84](#)



Nous Soutenir [👉](#)

Le blog contient des publicités, elles permettent de financer l'hébergement et maintenir le blog en fonctionnement. Vous pouvez utiliser adblock pour une lecture sans publicités.