Technique des microcontrôleurs

ETBK 3L f

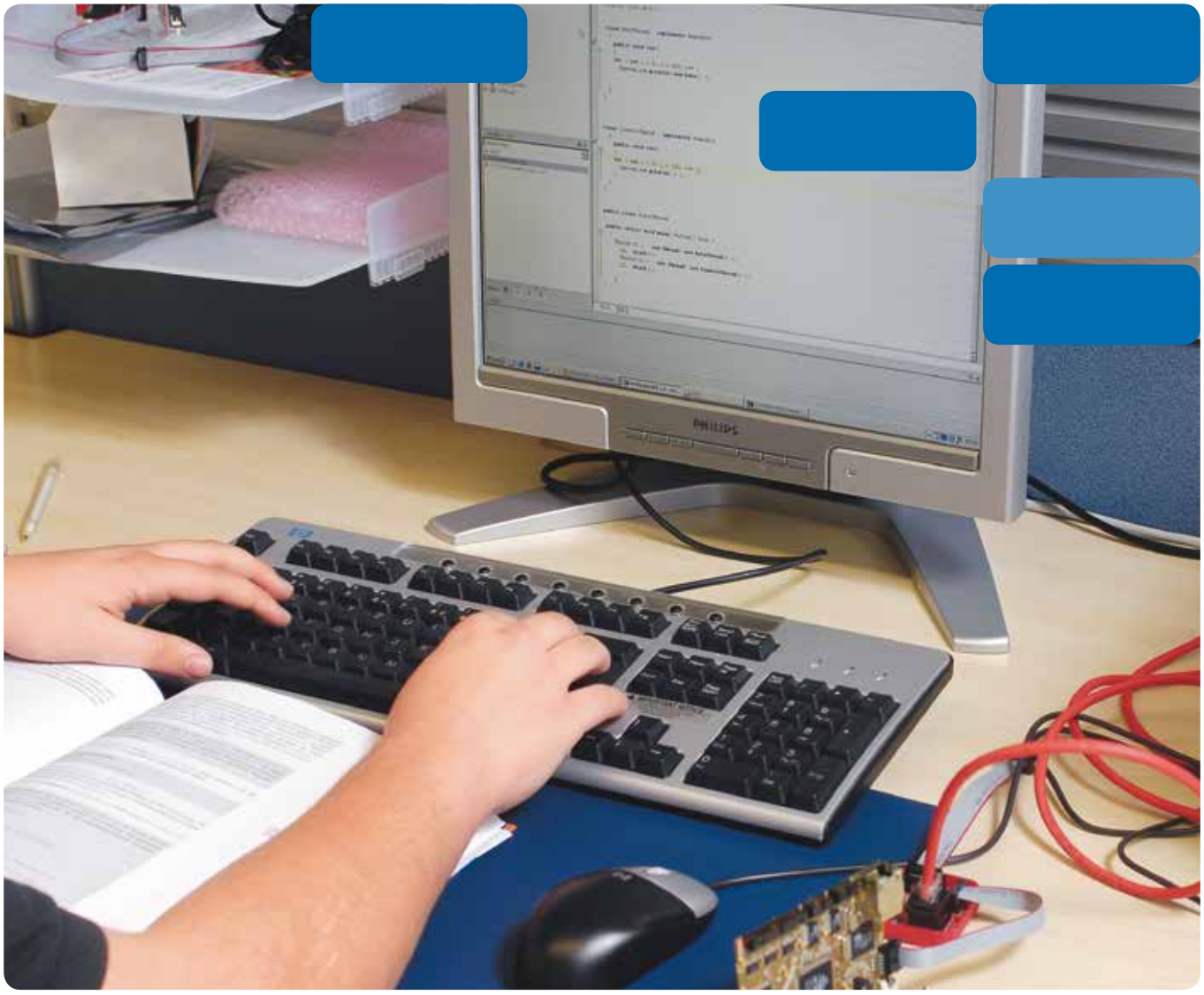


Table des matières

3 ETB3 Technique des microcontrôleurs
3.1 Compréhension des microcontrôleurs 7
3.2 Analyse et conception 105
3.3 Réalisation / Codage 133
3.4 Test du logiciel 141
3.5 Documentation 149
3.6 Présentation 167

ÉDITIONS SWISSMEM

SWISSMEM

Editeur: Editions Swissmem
5e édition 2018

Commandes:
Swissmem Berufsbildung
Brühlbergstrasse 4
8400 Winterthur

Téléphone service d'expédition 052 260 55 55
Fax service d'expédition 052 260 55 59

www.swissmem-berufsbildung.ch
vertrieb.berufsbildung@swissmem.ch

Copyright texte, dessins et graphisme:
© by Swissmem, Zurich

Tous droits réservés. L'œuvre avec toutes les parties qu'elle contient est protégée par les droits d'auteur. Toute utilisation dans d'autres cas que ceux prescrits par loi nécessite le consentement préalable écrit de l'éditeur.

SWISSMEM

Ont participé à l'élaboration de ce guide méthodique:

Baltisberger Martin, Müller Martini Druckverarbeitungssysteme, Zofingen
Bays Didier, CFPT, Petit-Lancy
Dupertuis Jean-Yves, Base aérienne de Payerne, Payerne
Habegger Olivier (chef de projet), Swissmem Berufsbildung, Winterthur
Inhelder Jürg, chef de projet de la 4e édition 2016, Winterthur
Liechti Simon, ICT Berufsbildungscenter, Bern
Manni Daniel, Bobst S.A., Mex
Schmied Toni, CPLN, Neuchâtel

Mai 2018 Swissmem Formation professionnelle

Explication des pictogrammes et structure du contenu

Explication des pictogrammes



Solution adéquate, dans le sens de l'optimisation d'un produit.



Solution utilisable. Il y a sûrement de meilleures solutions.



Solution inadéquate. Identifiez les points faibles et cherchez une meilleure variante.



Devoir. Utilisez les meilleurs moyens pour le résoudre.



Objectifs de formation



Remarque importante



Information



Informations sur le web: www.swissmem-elearning.ch

Notez ici les informations pertinentes, comme les normes nationales ou internationales, les normes de l'entreprise, les titres des ouvrages spécialisés, les modes d'emploi, etc.

Structuration du contenu

Le guide méthodique est structuré de la même manière que le catalogue des compétences-ressources.

Les ressources sont structurées de la manière suivante:

Activation
Chaque unité de formation commence par des questions de base qui vous permettent de contrôler votre niveau de connaissances actuel.

Théorie / Devoirs
Cette partie contient en plus de la théorie des questions et/ou des devoirs que les apprentis doivent résoudre.

Révision
Des questions de révision en fin de module permettent de vérifier l'acquisition de la ressource et de consolider la matière enseignée.

Table des matières

3.1 Compréhension des microcontrôleurs

3.1.0 Les bases de la technique binaire **7**
Représentation de l'information 8
Logique binaire – masquage 15
Arithmétique binaire 18
3.1.1 Evaluer les microcontrôleurs **27**
Qu'est-ce qu'un microcontrôleur? 28
Structure interne d'un microcontrôleur 29
Architecture du système 31
Modules mémoire 35
Commande et adressage 38
La programmation d'un microcontrôleur 42
3.1.2 Appliquer un langage de programmation évolué **45**
Langages de programmation 46
Langages évolués 48
Initiation au langage évolué C 51
Opérateurs 52
Variables 53
Boucles 56
Branchements conditionnels 58
Fonctions 60
Premier programme ETS 63
3.1.3 Appliquer les outils de développement **65**
Introduction 66
Environnement de développement intégré (IDE) 68
Assembleur et langage évolué (compilateur) 70
Programmation directe 70
Linker (éditeur de liens) 71
Déroulement global de la réalisation d'un projet 72
Exemples de fichier 73
Présentation d'un IDE 76
3.1.4 Ingénierie logicielle **81**
Conception de logiciels: du cahier des charges au test 82
Exercice 84
Analyse et conception 89
Portée des données 91
Données locales 91
Déclaration de données 92
Documentation 95
3.1.5 Mettre en service et contrôler des systèmes à microcontrôleurs **101**
Interpréter des fiches techniques 102
Ecrire des programmes de test pour des systèmes à microcontrôleurs 103
Analyser des tâches 105

Table des matières

3.2 Analyse et conception

3.2.1 Se documenter **105**
3.2.2 Concepts matériels **106**
3.2.3 Etablir la représentation graphique **107**
Structogramme «Nassi-Shneiderman» 108
Ordinogramme/ Diagramme de flux 115
Comparaison entre le structogramme et l'ordinogramme 118
Diagramme d'états-transitions, State-Event Diagram 123
Conception d'un automate à états finis 124
Propositions de mise en œuvre pour certaines fonctions du programme 127
Interrupteur, piloté en fonction de la position 127
Interrupteur, piloté par transition 128
Clignotement 130
Programme cadencé 131

3.3 Réalisation / Codage

3.3.1 Configurer et utiliser le compilateur et le débogueur **134**
Commentaires dans le code source C 134
Verbalisation 136
Codage en C 136

3.4 Test du logiciel

3.4.1 Effectuer les tests, les consigner dans un protocole **143**
Exemples 144

3.5 Documentation

3.5.1 Etablir la documentation du programme **151**
Cahier des charges 152
3.5.2 Projet poulailler **152**
Planification du projet 154
Analyse détaillée du projet 155
Diagramme de commande 156
Diagramme d'états-transitions 158
Table états-transitions 158
Structogramme 159
Codage 160
Test du programme 163

3.6 Présentation

3.6.1 Présenter le programme **167**

3.1.0 Les bases de la technique binaire

Comprendre les bases de la technique binaire

1) Enumérez les différentes bases arithmétiques couramment utilisées en électronique:

Binaire (base 2), décimal (base 10), hexadécimal (base 16) et éventuellement octal (base 8).

2) Dans quels domaines utilise-t-on la technique binaire?

Dans tous les systèmes numériques (processeurs d'ordinateurs, systèmes logiques, etc.), stockages et transmissions de données

3) Citez quelques exemples de représentation des états binaires:

«0» et «1», «faux» et «vrai», «bas» et «haut», «low» et «high»

4) Donnez quelques exemples de supports utilisant la technique binaire pour stocker des informations:

CD-ROM, DVD, lecteur mp3, disque dur d'ordinateur, clés USB

5) Comment pourrait-on représenter une information binaire en électronique?

En choisissant 2 potentiels différents pour représenter les 2 états logiques (p.ex. le «0» avec un potentiel à 0 [V] et le «1» avec un potentiel à 5 [V])

3.1.0 Les bases de la technique binaire

Représentation de l'information

Les termes numérique et analogique

La plupart des grandeurs existantes dans la nature (luminosité, température, pression, vitesse...) se situent dans une plage de valeurs donnée. On les qualifie de grandeurs **analogiques**. Elles peuvent être appréhendées par l'intermédiaire de circuits. De simples régulateurs de luminosité, ou encore des régulateurs de tension dans des dispositifs d'alimentation peuvent fonctionner de cette manière. Les grandeurs analogiques ne peuvent être traitées par les ordinateurs que sous forme numérique. Elles doivent donc être transformées en grandeurs **numériques** avant d'être transmises à l'ordinateur.

Des procédures de numérisation simples sont par exemple la répartition des:
– nombres réels en intervalles entiers positifs et négatifs
– le temps en heures, minutes, secondes
– la température en Kelvin, etc.



Aujourd'hui, la musique est le plus souvent enregistrée et vendue sous forme numérique. En format MP3 de la Fraunhofer-Gesellschaft, une minute de musique occupe 1,4 mégaoctet.

Si la valeur des intervalles entre ces valeurs discrètes est assez petite, nos sens humains ne remarquent plus que la numérisation a eu lieu. Cependant, cette répartition de plus en plus fine exige des capacités informatiques de plus en plus grandes. D'autant plus qu'un ordinateur ne peut traiter que des valeurs binaires, c'est-à-dire présentant deux états possibles. Les valeurs numériques doivent donc être codées avec plusieurs positions binaires. Quelques exemples des vitesses de traitement nécessaires montrent que même les microprocesseurs modernes risquent d'être rapidement dépassés:

– régulation de chauffage	10 valeurs binaires/s
– saisie manuelle sur un clavier	100 b/s
– transmission de la voix	100 kb/s
– positionnement de machine-outil	1 Mb/s
– traitement d'image couleur	10 Mb/s
– transmission HDTV	1,4…3,4 Mb/s
– transmission UHDTV (4K)	20...30 Mb/s

Des systèmes numériques ou même binaires ne se trouvent dans la nature que par définition, c'est-à-dire que nous attribuons à des processus naturels de manière arbitraire un attribut «oui/non». Dans la vie courante, un «oui/non» bien clair peut se transformer de manière imagée en une expression «un peu moins binaire»:
– Nuit ou jour (crépuscule, nuit noire)
– Froid ou chaud (froid glacial, chaleur torride)
– Mort ou vivant (raide mort, très actif)

3.1.0 Les bases de la technique binaire

Le bit

La quantité d'information peut être mesurée, l'unité de mesure s'appelle le bit. Un bit représente la plus petite quantité d'information (réponse à une question qui n'admet que deux réponses – oui ou non).

Le terme bit est l'abréviation de «**b**inary digi**t**» (chiffre binaire). Dans la pratique, il est également utilisé pour les signaux binaires et les états binaires.

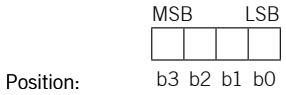
Les mots de données

Plusieurs bits peuvent être regroupés sous la forme d'un mot de données. Les bits sont numérotés, le bit de poids le plus faible LSB (**l**east **s**ignificant **b**it) reçoit le numéro zéro et est représenté à droite. Le bit de poids le plus fort est appelé MSB (**m**ost **s**ignificant **b**it).

Le nibble

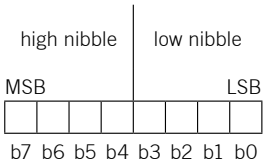
Un nibble est constitué de quatre bits, et forme donc un demi-octet (anglais «to nibble» = grignoter).

Les autres termes utilisés pour nibble sont:
– tétrade («latin tetra = quatre»)
– demi-octet



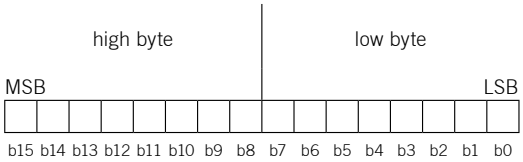
L'octet (Byte)

Un octet est constitué de huit bits et donc de deux nibbles de chacun quatre bits.



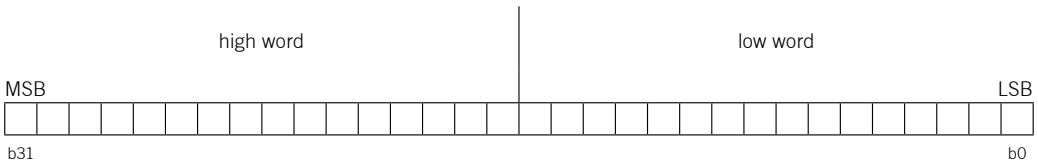
Le mot (Word)

Un mot est constitué de deux octets.



Le double mot (Long Word)

De nombreux micro-ordinateurs peuvent traiter des mots de données de 32 bits = 2 mots.



3.1.0 Les bases de la technique binaire

Désignation des valeurs significatives

Les désignations suivantes sont utilisées pour les valeurs significatives:

– LSB (least significant bit)	bit de poids le plus faible
– Low Bit	bit de poids faible
– Low Word	mot de poids faible
– MSB (most significant bit)	bit de poids le plus fort
– High Bit	bit de poids fort
– High Word	mot de poids fort

Notation

Lorsque l'on représente 8, 16 voire 32 bits, la suite de zéros et de uns est difficile à interpréter. C'est pourquoi on note souvent les mots de données sous format hexadécimal. C'est ainsi que, par exemple, l'octet représenté en notation binaire par 0011 1011 s'écrit en hexadécimal 3B. A l'aide de préfixes ou de suffixes (par exemple 3Bh), il est possible de passer, lors du développement de programmes sur les systèmes à micro-ordinateurs, d'une forme de saisie et d'affichage à une autre.

L'introduction de plusieurs systèmes de numération peut prêter à confusion dans l'interprétation des nombres. Pour distinguer les systèmes de notation, une convention a été adoptée pour la notation des nombres en ajoutant des préfixes ou des suffixes. Un nombre sans préfixe ni suffixe est écrit en décimal.

Notation	**Représentation**
Binaire	10B ou 10b ou 10_b ou %10 ou $10_{(2)}$
Décimal	10D ou 10d ou 10_d ou 10 ou $10_{(10)}$
Hexadécimal	10H ou 10h ou 10_h ou 0x10 ou $10 ou $10_{(16)}$

Codes
Introduction

Les ordinateurs et autres circuits de traitement numériques peuvent uniquement traiter que des signaux binaires. Les valeurs de l'environnement réel (signaux analogiques, caractères d'impression, etc.) doivent être représentées sous forme de valeurs numériques en mots de données.

La convention qui permet d'affecter à chaque élément d'une grandeur (par exemple valeur de mesure) un élément d'une autre grandeur (par exemple valeur numérique) est appelée code.

Pour pouvoir interpréter un ensemble de bits dans l'ordinateur, on doit connaître le code du mot de données concerné. Des ensembles de bits identiques peuvent, suivant le code, avoir une signification complètement différente.
Dans les codes pondérés, chaque bit individuel représente en lui même une valeur donnée. Les codes non pondérés ne permettent d'interpréter que l'ensemble du mot de données.

Codes non pondérés

Les codes de caractères sont un exemple typique de codes non pondérés. Lorsqu'un ordinateur traite des textes (enregistrement, impression, etc.), il manipule en interne chaque caractère sous forme d'un mot de données. Selon le fabricant de matériel, la langue, etc., différents codes sont utilisés.

Le code «**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange» est le code le plus répandu pour traiter des textes. Il est issu du code télex américain et utilise des mots de données de 7 bits. Les codes étendus utilisent partiellement huit bits ou définissent des caractères supplémentaires suivant les pays ou les applications.

3.1.0 Les bases de la technique binaire

Code ASCII

100 1101 ⇔ M
110 0001 ⇔ a
101 1011 ⇔ [
010 0001 ⇔ !

Table ASCII standard

Dec	Hex	Caract. de contrôle / Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End transm. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

SWISSMEM

3.1.0 Les bases de la technique binaire

Table ASCII étendue

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	└	224	E0	α
129	81	ü	161	A1	í	193	C1	┴	225	E1	β
130	82	é	162	A2	ó	194	C2	┬	226	E2	Γ
131	83	â	163	A3	ú	195	C3	├	227	E3	π
132	84	ä	164	A4	ñ	196	C4	─	228	E4	∑
133	85	à	165	A5	Ñ	197	C5	┼	229	E5	σ
134	86	å	166	A6	ª	198	C6	╞	230	6E	µ
135	87	ç	167	A7	º	199	C7	╟	231	E7	τ
136	88	ê	168	A8	¿	200	C8	╚	232	E8	Φ
137	89	ë	169	A9	⌐	201	C9	╔	233	E9	Θ
138	8A	è	170	AA	¬	202	CA	╩	234	EA	Ω
139	8B	ï	171	AB	½	203	CB	╦	235	EB	δ
140	8C	î	172	AC	¼	204	CC	╠	236	EC	∞
141	8D	ì	173	AD	¡	205	CD	═	237	ED	π
142	8E	Ä	174	AE	«	206	CE	╬	238	EE	ε
143	8F	Å	175	AF	»	207	CF	╧	239	EF	∩
144	90	É	176	B0	░	208	D0	╨	240	F0	≡
145	91	æ	177	B1	▒	209	D1	╤	241	F1	±
146	92	Æ	178	B2	▓	210	D2	╥	242	F2	≥
147	93	ô	179	B3	│	211	D3	╙	243	F3	≤
148	94	ö	180	B4	┤	212	D4	╘	244	F4	⌠
149	95	ò	181	B5	╡	213	D5	╒	245	F5	⌡
150	96	û	182	B6	╢	214	D6	╓	256	F6	÷
151	97	ù	183	B7	╖	215	D7	╫	247	F7	≈
152	98	ÿ	184	B8	╕	216	D8	╪	248	F8	▫
153	99	Ö	185	B9	╣	217	D9	┘	249	F9	▪
154	9A	Ü	186	BA	║	218	DA	┌	250	FA	·
155	9B	¢	187	BB	╗	219	DB	█	251	FB	√
156	9C	£	188	BC	╝	220	DC	▄	252	FC	n
157	9D	¥	189	BD	╜	221	DD	▌	253	FD	2
158	9E	₧	190	BE	╛	222	DE	▐	254	FE	■
159	9F	ƒ	191	BF	┐	223	DF	▀	255	FF	□