

Table des matières

Avant-propos	1
Chapitre 1. Objectifs, contexte et exemple fil rouge	3
1.1. Objectifs de l'ouvrage	3
1.2. Ce que contient l'ouvrage... et ce qu'il ne contient pas	4
1.3. À propos des exemples de programmes	5
1.4. Choisir son μ contrôleur : la famille STM32	6
1.4.1. Critères de choix d'un μ contrôleur	6
1.4.2. La famille STM32	7
1.4.3. La ligne STM32F10x	8
1.5. Exemple fil rouge	9
1.5.1. Le contexte : installation photovoltaïque en autoconsommation	11
1.5.2. Le système : optimisation du chauffage électrique dans un foyer en autoconsommation	11
1.5.2.1. L'analyseur de puissance	14
1.5.2.2. L'unité centrale	14
1.5.2.3. Les boîtiers de télécommande	16
1.5.3. Synthèse des périphériques vus à titre d'exemple fil rouge	16
Chapitre 2. Principes généraux de programmation	19
2.1. Pointeurs et registres	19
2.1.1. La variable « simple »	19
2.1.2. Le pointeur sur une variable	20
2.1.3. Utilisation de pointeurs pour le passage d'arguments à une fonction	22

2.1.4. Quand utiliser un pointeur ?	25
2.1.5. Le pointeur de fonction	25
2.2. Structure en langage C et usage pour le STM32	26
2.2.1. Organisation en type structuré	26
2.2.2. Accès aux registres <i>via</i> une structure	28
2.3. Cas des structures imbriquées	29
2.4. L'utilisation des masques	31
2.4.1. Considérations générales	32
2.4.2. Masque de mise à un	33
2.4.3. Masque de mise à zéro	35
2.4.4. Inverser un bit	36
2.4.5. Extension et généralisation à l'affectation d'un champ de bits	36
2.5. De la structuration du code et la notion de pilote	37
2.6. Les routines d'interruption	41
2.6.1. Notion de routine de traitement	41
2.6.2. Mise en œuvre des routines d'interruption	41
2.6.3. Redirection par programmation dynamique	44

Chapitre 3. Considérations matérielles générales des STM32F10x 47

3.1. L'architecture globale d'un STM32	47
3.2. Les composantes du Cortex-M3	49
3.3. L'arborescence des horloges et l'unité RCC	51
3.3.1. Génération des horloges primaires	51
3.3.2. Génération des horloges secondaires	52
3.3.3. Programmation de l'unité RCC	54
3.3.4. Exemple de codage d'une configuration de l'unité RCC	60
3.4. Les <i>watchdogs</i>	61
3.4.1. Le <i>watchdog</i> indépendant	62
3.4.2. Le <i>watchdog</i> fenêtré	64

Chapitre 4. Les entrées/sorties (E/S) binaires : port parallèle 65

4.1. La notion de port d'E/S	65
4.2. L'aspect électronique	67
4.2.1. Architecture globale	67
4.2.2. Configuration en entrée	69
4.2.3. Configuration en sortie	71
4.3. Les GPIO du STM32	73
4.3.1. Les registres de configuration	73
4.3.2. Les registres d'accès	75

4.4. Exemple	78
4.4.1. Proposition de solution	81
4.4.2. Commentaires	86
4.5. Les points non abordés concernant les GPIO	86
Chapitre 5. Gestion des interruptions et de la DMA	89
5.1. Les exceptions en général, les interruptions en particulier	89
5.2. Le <i>reset</i> et la table des vecteurs d'interruption	90
5.2.1. Les pointeurs de pile (SP) et d'instruction (PC)	91
5.2.2. Le contenu d'une table des vecteurs d'interruption	92
5.3. Le principe du détournement et le rôle du NVIC	94
5.3.1. Le NVIC	94
5.3.1.1. Son rôle	95
5.3.1.2. Ses principaux registres	95
5.3.2. Les conditions pour qu'une interruption soit acceptée	97
5.3.3. La gestion des priorités	98
5.3.4. Les étapes du détournement	102
5.4. Le cas des entrées d'interruption externe	103
5.4.1. Le périphérique AFIO et les EXTI	104
5.4.2. Le niveau NVIC	105
5.4.3. Les niveaux GPIO et AFIO	106
5.4.4. Le niveau EXTI	109
5.5. Exemple fil rouge : traitement d'une exception EXTI	110
5.5.1. Le problème à résoudre	110
5.5.2. Proposition de solution	110
5.6. L'accès direct en mémoire (DMA)	112
5.6.1. Généralités	112
5.6.2. La DMA sur le STM32F10x	114
5.7. Exemple fil rouge : cas d'une programmation de DMA	119
5.7.1. Configuration de l'UART	120
5.7.2. Configuration de la DMA	120
5.7.3. Structure du programme	120
Chapitre 6. Les unités Timers	123
6.1. Généralités sur les unités de comptage ou de Timer	123
6.1.1. Principe de base	124
6.1.2. Les unités Timers du STM32F10x	125
6.1.3. Les différents types de Timers	125
6.2. Les modes de fonctionnement	127
6.2.1. La structure générale d'un Timer	127

6.2.2. Le bloc <i>Core</i> : débordement, événement et fréquence	128
6.2.2.1. Le débordement	129
6.2.2.2. Comptage et décomptage	130
6.2.2.3. L'événement <i>Update</i> (UEV)	131
6.2.3. Le bloc <i>Capture/Compare</i>	133
6.2.3.1. Le mode <i>Capture</i>	133
6.2.3.2. Le mode <i>Compare</i>	136
6.2.4. Le bloc contrôleur <i>Trigger</i> et interface codeur	139
6.2.4.1. Choix de l'horloge du cœur du Timer	140
6.2.4.2. Le mode <i>Slave</i>	142
6.2.4.3. Le mode <i>One Pulse</i>	144
6.2.4.4. Le mode codeur incrémental	146
6.2.5. Le mode <i>Master</i>	148
6.3. Un Timer supplémentaire : le SysTick	149
6.4. Exemple fil rouge	152
6.4.1. Cas d'étude : molette de commande	152
6.4.2. Extrait de code	152

Chapitre 7. Modulation en largeur d'impulsion (PWM) 155

7.1. Généralités sur les signaux de type PWM	155
7.1.1. Définition	155
7.1.2. Utilisation en transmission d'information	157
7.1.3. Utilisation en conversion numérique/analogique	158
7.1.3.1. Application à la création d'une tension analogique équivalente	158
7.1.3.2. Application à la commande de moteur	163
7.2. Mise en œuvre de la PWM dans un μ contrôleur STM32F10x	164
7.2.1. Fonctionnement de base	164
7.2.2. Configuration et options	165
7.3. Exemple fil rouge	170
7.3.1. Première solution envisageable	170
7.3.2. Seconde solution retenue	171
7.3.3. Configuration des interruptions	171
7.3.4. Configuration de la PWM en elle-même	171
7.3.5. Configuration du GPIO	172
7.3.6. Extrait du code	172

Chapitre 8. Convertisseur analogique numérique 175

8.1. Généralités et principes de fonctionnement	175
8.1.1. Rôle et définitions	175
8.1.2. ADC à approximations successives	177

8.1.3. La mesure de tension variable	179
8.1.3.1. Approche idéalisée	179
8.1.3.2. Approche réelle	180
8.1.4. ADC à plusieurs voies d'entrée	182
8.1.5. Point sur les timings de l'ADC	183
8.2. L'ADC sur le STM32F10x	184
8.2.1. Architecture simplifiée de l'ADC	184
8.2.1.1. Le multiplexage et l'échantillonnage des voies	185
8.2.1.2. Le déclenchement de l'ADC	188
8.2.1.3. Le résultat de conversion	189
8.2.1.4. La calibration de l'ADC	190
8.2.1.5. Les interruptions de l'ADC	190
8.2.1.6. Les réglages obligatoires indépendamment du mode choisi	191
8.2.2. Les modes de fonctionnement de l'ADC	192
8.2.2.1. Mode conversion sur voie unique, simple ou continu	193
8.2.2.2. Mode conversion balayage sur voies multiples, simple ou continu	194
8.2.2.3. Mode discontinu	195
8.2.3. La DMA avec l'ADC	196
8.3. Application : l'analyseur de puissance	197
8.3.1. Détermination des durées d'échantillonnage T_s	197
8.3.2. Extrait de code de configuration	198

Chapitre 9. Quelques bus de communication 201

9.1. Introduction	201
9.2. Les unités UART	203
9.2.1. Généralités	203
9.2.2. Techniques de lecture/écriture	206
9.2.2.1. Émission par l'UART	206
9.2.2.2. Réception par l'UART	207
9.2.3. Configuration de l'UART sur les STM32F10x	207
9.2.3.1. Le registre de réception DR	208
9.2.3.2. Différents <i>flags</i> utiles dans le SR	208
9.2.3.3. Le contrôle de la vitesse de transmission dans le BRR	210
9.2.3.4. Les registres de configuration CR1, CR2 et CR3	211
9.3. Les unités SPI	214
9.3.1. Généralités	214
9.3.2. Techniques de lecture/écriture basées sur un registre à décalage	216
9.3.2.1. Principe d'un registre à décalage	216
9.3.2.2. Techniques de lecture/écriture	219
9.3.3. Configuration du SPI des STM32F10x	220
9.3.3.1. Le registre de données DR	221

- 9.3.3.2. Les registres de configuration CR1, CR2 221
- 9.3.3.3. Différents *flags* utiles du registre SR 223
- 9.4. Le bus I²C 224
 - 9.4.1. Généralités 226
 - 9.4.2. Topologie électrique du bus I²C 226
 - 9.4.3. Principe et protocole 228
 - 9.4.3.1. La transmission d'un bit 228
 - 9.4.3.2. La délimitation de trame 228
 - 9.4.3.3. L'acquittement (*Acknowledge*) 228
 - 9.4.3.4. L'adressage I²C 231
 - 9.4.4. Techniques de lecture/écriture 231
 - 9.4.4.1. Émission I²C 232
 - 9.4.4.2. Réception I²C 232
 - 9.4.5. Configuration de l'I²C sur le STM32F10x 235
 - 9.4.5.1. La configuration générale du périphérique I²C 235
 - 9.4.5.2. Le démarrage d'une communication (*start, I²C Slave @, Word @*) 239
 - 9.4.5.3. Fermeture d'une communication 239
 - 9.4.5.4. Solution par scrutation : écriture d'un bloc de données 240
 - 9.4.5.5. Solution par scrutation : lecture d'un bloc de données 241
 - 9.4.5.6. Solution avec DMA : configuration supplémentaire 242
 - 9.4.5.7. Solution avec DMA : écriture d'un bloc de données 242
 - 9.4.5.8. Solution avec DMA : lecture d'un bloc de données 244
- 9.5. L'exemple fil rouge 245
 - 9.5.1. Réception UART en interruption 245
 - 9.5.1.1. Problème à résoudre 245
 - 9.5.1.2. Proposition de solution 245
 - 9.5.2. Mesure de température par bus I²C et DMA 246
 - 9.5.2.1. Problème à résoudre 246
 - 9.5.2.2. Proposition de solution 247
 - 9.5.2.3. Le fichier Driver_I2C.h 247
 - 9.5.2.4. Extrait du fichier Principal.c 248
 - 9.5.2.5. Le fichier Driver_I2C.c 248

Chapitre 10. Gestion de l'énergie sur STM32 253

- 10.1. Introduction 253
- 10.2. Les domaines du μ contrôleur 254
 - 10.2.1. *1.8V Domain* 254
 - 10.2.2. *V_{DD} Domain* 256
 - 10.2.3. *Backup Domain* 256
 - 10.2.4. *V_{DDA} Domain* 257

10.3. <i>Reset</i> sur STM32	257
10.3.1. <i>System Reset</i>	257
10.3.2. <i>Power Reset</i>	259
10.3.3. <i>Backup Domain Reset</i>	259
10.3.4. Gestion des <i>reset flags</i> (RCC_CSR)	260
10.3.5. Résumé des <i>resets</i> sur STM32F10x	261
10.4. La RTC	262
10.4.1. Accès sécurisé au <i>Backup Domain</i>	262
10.4.2. Le registre RCC_BDCR	262
10.4.3. Constitution de la RTC	263
10.4.4. Lecture/écriture des registres RTC	265
10.4.4.1. Lecture	265
10.4.4.2. Écriture	266
10.5. Les modes <i>Low Power</i>	267
10.5.1. Événement et interruption	267
10.5.2. Les bits importants pour les modes <i>Low Power</i>	268
10.5.2.1. Les bits du Cortex-M3	268
10.5.2.2. Les bits externes au Cortex	270
10.5.3. Le mode <i>Sleep</i>	271
10.5.3.1. Entrée dans le mode <i>Sleep</i>	271
10.5.3.2. Sortie du mode <i>Sleep</i>	272
10.5.4. Le mode <i>Stop</i>	272
10.5.4.1. Entrée dans le mode <i>Stop</i>	272
10.5.4.2. Sortie du mode <i>Stop</i>	273
10.5.5. Le mode <i>Standby</i>	273
10.5.5.1. Entrée dans le mode <i>Standby</i>	273
10.5.5.2. Sortie du mode <i>Standby</i>	273
10.6. Architecture logicielle et utilisation des modes <i>Low Power</i>	274
10.6.1. Architecture logicielle classique	274
10.6.2. Mise en sommeil dans le <code>main()</code> , instruction <code>WFI</code>	275
10.6.3. Mise en sommeil automatique lors de la sortie d'interruption	275
10.6.4. Mise en sommeil dans le <code>main()</code> , instruction <code>WFE</code>	275
10.6.5. Mode <i>Standby</i> sur alarme RTC	277
10.7. Exemple fil rouge pour le mode <i>Low Power</i>	277
10.7.1. Problème à résoudre	277
10.7.2. Proposition de solution	279

Annexe. Éléments technologiques	283
--	------------

Bibliographie	295
--------------------------------	------------

Index	297
------------------------	------------