

**SYSTEMES LOGIQUES
PROJET DE FIN DE SEMESTRE****1. DÉROULEMENT DU PROJET**

Un projet est organisé dans les dernières semaines du semestre. Le but du projet est de mettre en pratique les compétences acquises dans les parties théoriques et au cours des travaux pratiques dans le contexte d'un projet complexe impliquant des contraintes proches d'un produit commercial simple.

Le projet peut être débuté dès la réception du sujet, mais au plus tard dès la fin du TP04. Le projet est effectué par groupes de deux étudiants. Les salles de TPs ne sont à disposition en dehors des heures de cours uniquement s'il n'y a pas cours, et dans le cadre des directives EPFL relatives au COVID-19. Il est recommandé d'installer les logiciels sur des ordinateurs privés afin de travailler avec des horaires souples. Le descriptif de l'installation est fourni sur le site Moodle du cours; un support supplémentaire à l'installation n'est pas possible.

2. SUJET

Le projet consiste dans le développement d'une montre digitale avec fonctions annexes, qui soit contrôlée par un nombre minimal de boutons (deux à trois boutons-poussoirs, DIP-Switches), et affiche sur un affichage limité (six afficheurs à 7-segments, LEDs). L'utilisation du périphérique buzzer piézo-électrique est optionnelle. L'utilisation d'une carte périphérique 4x4 keypad est obligatoire.

Une exemple est proposé en annexe, présentant une montre commerciale avec des fonctions annexes.

Le cœur du problème réside en le fait que le nombre de boutons-poussoirs nécessaires au contrôle soit limité (deux ou trois, plus le banc de DIP-Switches) et l'affichage soit limité (six caractères). La solution réside dans le développement de plusieurs modes d'opérations qui constituent la machine d'états finis générale du système. La machine d'état générale doit être développée suivant la méthode présentée au cours, et les DIP-switches ou les boutons du keypad ne doivent pas servir à contrôler/générer la machine d'états principale (FSM générale).

Méthode de développement proposée

- effectuez une étude sur papier afin d'identifier l'architecture du projet (les blocs et leur interconnexions) ainsi que les spécifications (fréquence minimale, etc), puis
- développez une machine d'états finis générale simple, ainsi que quelques modules simples (heure, chronomètre) qui lui sont rattachés comme sous-modules et validez votre méthode de travail par un test sur la carte; puis,
- développez des modules plus complexes indépendamment, validez-les, puis seulement à ce moment rattachez-les à la FSM générale.

3. DÉLIVRABLES

Trois livrables sont requis, qui doivent être téléchargés sur le site Moodle du cours, au plus tard le 13.12.2021, à 08:00. Un seul étudiant par groupe fournit les livrables. Les trois documents sont les suivants:

- un fichier .circ développé sous logisim-evolution et qui contient la totalité du projet, dans sa dernière version parfaitement fonctionnelle sur la carte DE10-Lite;
- une vidéo de démonstration succincte (3 minutes maximum, 50 MB);
- un rapport concis de trois pages, format A4.

Le rapport sera organisé de la façon suivante. Une première page sert à la description générale du projet, et à un mode d'emploi. Les deux pages suivantes présentent les solutions techniques apportées, comprenant une description de la machine d'états finis générale, des machines d'états secondaires spécialement intéressantes (incluant la solution à l'utilisation du périphérique), ainsi que une description de solutions techniques originales apportées à la résolution de problèmes rencontrés dans le développement; présentez des schémas répliqués de logisim-evolution, dans un format synthétique dont la lisibilité soit adaptée à un rapport, des diagrammes temporels, des schémas-blocs ou diagrammes de flux, respectivement les mieux adaptés à supporter vos explications.

4. PRÉSENTATION DU PROJET

La démonstration du projet se fait par le moyen d'une vidéo brève de trois minutes au maximum présentant les diverses fonctionnalités réalisées. La vidéo peut être réalisée par des moyens légers (téléphone cellulaire, par exemple) et aucun montage n'est nécessaire. Il est important par contre que les fonctionnalités soient démontrées et expliquées et il est donc important que la présentation soit très bien préparée et organisée. Il n'y a pas de défense à proprement parler, consistant en une argumentation des solutions apportées; le rapport sera utilisé à cette fin.

5. RENDU DU MATÉRIEL

Les cartes DE10-Lite et périphériques seront collectés selon l'horaire suivant:

- Lundi 13.12.2021 en CO-2 de 15:15 à 17:00;
- Jeudi 16.12.2021 en MED 2-2519 de 15:00 à 17:00.

La totalité du matériel en prêt doit être rendu à cette occasion, et les pannes doivent être annoncées. Aucune note ne sera donnée au cours sans rendu de la carte. Une alternative sera préparée et placée sur le site Moodle si les conditions sanitaires ne permettent pas la remise aux dates ci-dessus.

6. NOTATION

La notation du projet sera basée sur les éléments suivants

- fonctionnalité du projet,
- rapport (qualité, précision des explications),
- développement modulaire et hiérarchique,
- présentation (vidée claire et préparée).

L'extension des fonctionnalités implémentées permettra de déterminer une note cible.

- Fonction implémentée parfaitement fonctionnelle: gestion des fonctionnalités de base de la montre (heure et affichage, set d'une heure de réveil et alarme sonore/visuelle); note maximale: 5.0.
- Fonctionnalités additionnelles (deux fonctions simples): chronomètre, countdown, fuseaux horaires, etc.; note maximale: 5.5.
- Fonctionnalités additionnelles (plus de deux fonctions, complexes): fonctions originales complexes supplémentaires (calculatrice, pseudo-graphique d'animation des LEDs, programmation des messages défilants, animation de LEDs 7-segment, accès à des périphériques supplémentaires, jeu, musique, etc.); note maximale 6.

Le non respect de l'utilisation de la carte périphérique entraine un retrait: pas de carte (keypad) ni buzzer: - 1.5 pt; uniquement buzzer utilisé: -1.0 pt.

7. RAPPEL: FORMATION DE LA NOTE FINALE

Selon l'information que vous avez reçue en début du semestre, également reportée sur la page web du cours (<https://moodle.epfl.ch/course/view.php?id=15423>), et en conformité avec l'Ordonnance générale sur le contrôle des études à l'Ecole polytechnique fédérale de Lausanne, la note finale est formée pour 30% du résultat du projet et pour 70% du résultat du contrôle écrit des connaissances.

8. ANNEXES

Annexe 1: Méthode proposée au développement du circuit de contrôle de la carte périphérique.

Annexe 2: Exemple de montre numérique commerciale implémentant des fonctionnalités de base (extrait de mode d'emploi) [2].

9. ANNEXE 1: CARTE PÉRIPHÉRIQUE

Une carte périphérique disposant d'un clavier 4x4 keypad. Une méthode possible pour le développement du circuit de contrôle est proposée dans la suite qui aboutit à une solution possible.

9.1 PARTIE MATÉRIELLE

La carte périphérique "4x4 keypad" et ses connexions sont présentées en Figure 9.1. De plus un périphérique buzzer est également présenté.

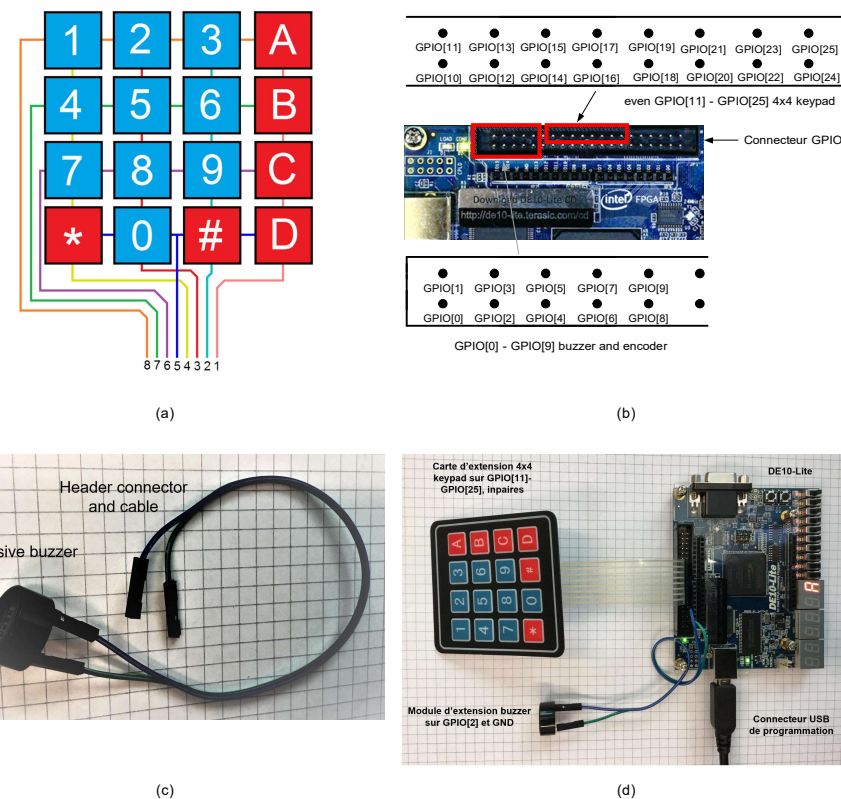


Figure 9.1: (a) Carte périphérique 4x4 keypad, vue du dessus et connexions, tiré de [1]. (b) Vue des connexions GPIO de la carte DE10-Lite. (c) photo du périphérique buzzer passif et connecteurs, et (d) DE10-Lite indiquant les connexions utilisées pour connecter la carte périphérique, et photo du système monté avec deux périphériques.

La carte 4x4 keypad (Figure 9.1(a)) est placée sur le port d'extension GPIO de la carte DE10-Lite. Les pins situées vers le milieu de la carte sont utilisées. Ces pins sont nommées GPIO[11] à GPIO[25] et seules les pins impaires, situées vers l'extérieur de la carte DE10-Lite sont utilisées, Figure 9.1(b).

Le buzzer est placé sur le connecteur d'extension GPIO de la carte DE10-Lite; les signaux sont VDD et un signal. Les pins situées vers le "haut" de la carte (côté du connecteur USB de programmation) sont utilisées. Ces pins sont nommées GPIO[0] à GPIO[9]. Elles ont été programmées en pull-up; une résistance interne à la FPGA sert à cet effet. Les autres pins du port GPIO ne sont pas utilisées dans cet application; par exemple, GPIO[10] = Vdd, GPIO[11] = GND, etc.

Ainsi, lors de l'utilisation du module "FPGA Synthesize and Download" (FPGA Commander) dans logisim-evolution, il faudra aussi configurer correctement les pins du port GPIO, afin que les signaux du circuit logique soient connectés correctement aux pins de la carte d'extension. L'oscilloscope peut être utilisé pour vérifier les bonnes connexions.

9.2 DÉVELOPPEMENT DU CIRCUIT DE CONTRÔLE

Le buzzer est constitué d'un haut-parleur piézo-électrique placé dans une cage de résonance en plastic. Un signal carré de fréquence contrôlée lui est transmis qui est transduit en son.

La carte 4x4 keypad consiste en une matrice d'interconnexion dont il faut gérer le contrôle afin de détecter laquelle parmi les seize touches a été activée. Conceptuellement, la détection de touche est effectuée en deux phases consécutives

dans une première phase, une détection de touche et une détection de la ligne dans laquelle cette touche se trouve sont effectuées.

Un circuit implémentant certaines parties importantes de la détection de touche est à disposition sur le site Moodle du cours, ee110-ee207_hexdddec_example.circ. Le circuit comprend les modules suivants

- Une machine d'état générale. Il s'agit d'une FSM de Moore qui comprend une logique de transition consistant en une simple porte OR, ainsi que le compteur d'états et le décodeur qui sont réunis en une logique optimisée et réalisée par quatre DFFs. A la sortie du décodeur, chacun des états consécutifs est activé à logic-1. A l'entrée de la logique de transition sont présentées quatre impulsions qui sont générées au terme de chacun des quatre états consécutifs. Une impulsion fait ainsi évoluer la FSM vers l'état suivant.
- La logique gérant l'état 0, module state_0. Dans ce cas, les quatre colonnes sont conduites à l'état logique-0 (x1, x2, x3, x4) alors que les quatre lignes en entrée (y1, y2, y3, y4) sont tirées à l'état logique-1 par des résistances de pull-up qui sont à l'intérieur des pads de la FPGA. Lorsqu'un bouton est appuyé, un chemin est créé qui tire une seule des lignes (y1, y2, y3, y4), à logic-0. Le flanc descendant ainsi généré au moment de l'appui sur le bouton est détecté par la logique située dans le module state_0. La valeur des lignes (y1, y2, y3, y4) est alors un code qui comprend trois logic-1 et un logic-0; ce dernier logic-0 identifie la ligne active. Ce code est stocké dans une mémoire (DFFs). Finalement, toutes les entrées des boutons sont filtrées afin d'éviter des glitches.
- La logique gérant l'état 1, module state_1_to_be_completed. Cette logique comprend trois blocs logiques distincts et gère les canaux de sortie (x1, x2, x3, x4). L'état 1 se déroule en deux phase (états ou sous-états) consécutifs. Dans un premier temps, lorsque la FSM générale est à l'état 0, alors l'état 1 est en reset est tire toutes les canaux de sorties (x1, x2, x3, x4) à logic-0 afin de permettre la détection de ligne appuyée part l'état 0. Lorsque la FSM générale active l'état 1, alors, les canaux de sorties (x1, x2, x3, x4) sont toutes mises à l'état logic-1. Puis, chaque ligne, l'une après l'autre est mise à l'état logique 0. Ainsi, lorsque une ligne (y1, y2, y3, y4) est détectée à logic-0 (détection du flanc descendant), on observe laquelle des colonnes (x1, x2, x3, x4) est à ce moment à logic-0. Ce code est

constitué de trois logic-1 et un logic-0; ce dernier logic-0 identifie la colonne active. Ce code est stocké dans une mémoire (DFFs). Ainsi, le premier module de l'état 1 détecte le flanc descendant, le deuxième module génère le signal qui active chaque ligne à zéro, une après l'autre et doit être complété (compteur à bulle), et le troisième module gère les signaux de set et reset des bascules DFF.

- La logique gérant l'état 2, module state_2. Une impulsion est générée au reset du circuit. Ce signal peut être utilisé afin d'utiliser les codes de deux fois quatre bits reçus.
- La logique gérant l'état 3, module state_3. La FSM reste dans l'état 3 jusqu'à l'overflow d'un compteur 8-bit. Les signaux générés par la fin de l'état 2 et la fin de l'état 3 sont combinés et servent à générer une impulsion sur une LED qui est déclenchée à l'appui d'une touche et qui soit suffisamment longue pour être visible; un son peu aussi être activé pendant ce temps.
- Les signaux qui sont reçus de la matrice 4x4 keypad doivent être décodés et envoyés à l'affichage 7-segments. Les signaux sont constitués de deux codes 4-bit, l'un décrivant la ligne active et l'autre la colonne activée, par exemple, 0b01111110.

La Figure 9.2 présente un exemple de détection de bouton appuyé.

- Figure 9.2(a), state_0, détection de ligne. X1=X2=X3=X4=logic-0 et on observe que Y2=logic-0. Le code stocké est donc 0b01011 (les valeurs de Yn)
- Figure 9.2(b), state_1, détection de colonne. X1=X2=X3=X4=logic-1, puis un après l'autre, chacun des Xm est mis à logic-0 jusqu'à ce que la détection soit effectuée; ici X1 est à logic-0, et Y1=Y2=Y3=Y4=logic-1. Pas de détection.
- Figure 9.2(c), Suite de state_1, ici X2 est à logic-0, et Y1=Y2=Y3=Y4=logic-1. Pas de détection.
- Figure 9.2(d), Suite de state_1, ici X3 est à logic-0, et Y2=logic-0; ainsi, la colonne X3 est détectée. Le deuxième code stocké est donc 0b1101 (les valeurs de Xm).

Finalement, la colonne X3 et Y2 sont détectés et le code 0b01111101 a été mis en mémoire. Ce code correspond à un affichage qui doit être émis vers un afficheur 7-segment.

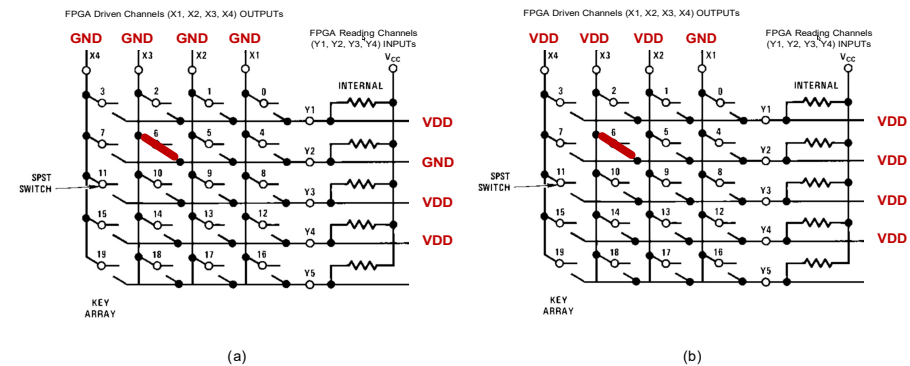


Figure 9.2: Opération du décodeur 4x4 keypad, tiré de [2]; (a) state_0, détection de ligne. (b), (c), (d) state_1 consécutifs jusqu'à la détection de colonne en (d).

RÉFÉRENCES

[1] 4x4 Matrix Membrane Keypad (#27899), Parallax Inc., v1.2.

[2] MM94C922, datasheet,

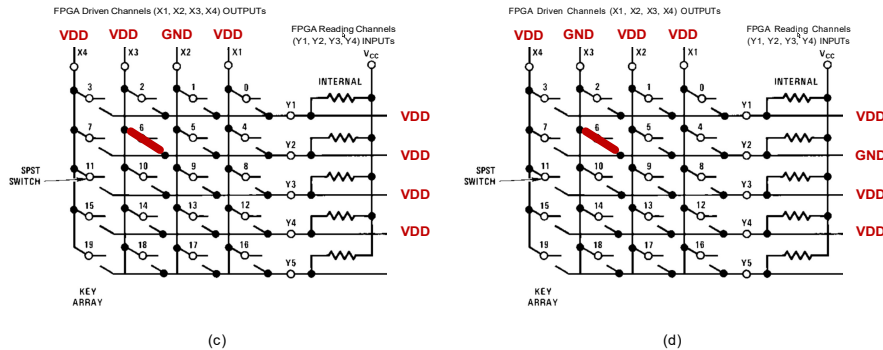


Figure 9.2: Opération du décodeur 4x4 keypad, tiré de [2]; (a) state_0, détection de ligne. (b), (c), (d) state_1 consécutifs jusqu'à la détection de colonne en (d).

Dans l'état courant, le circuit proposé réalise la machine d'états finis complète, les états state_0, state_2 et state_3, et tous les reset, détections de flancs et timings. Il est nécessaire de compléter l'état state_1 et le décodeur.

Ainsi, dans son état courant, le fichier fonctionne partiellement. Les lignes sont décodées et affichées de façon sommaire, tel que présenté en Figure 9.3. Puisque l'état 1 n'est pas terminé, il est nécessaire d'effectuer un reset à chaque nouvel appui d'une touche. Le fichier permet de débiter rapidement dans le projet en étant sûr du bon fonctionnement du matériel. Deux modules doivent être complétés.

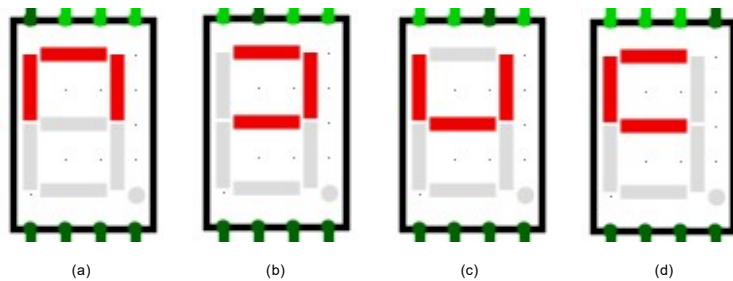


Figure 9.3: Différentes lignes détectées.

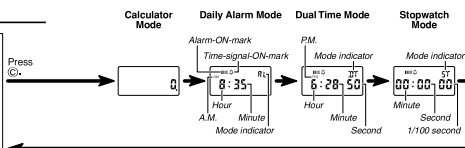
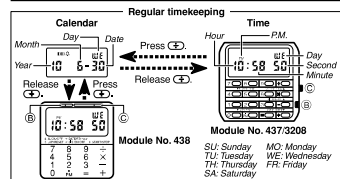
Les trois tâches suivantes devraient être effectuées au plus tôt dans le projet.

- comprendre la méthode de détection, et le fonctionnement des circuits existants;
- terminer l'état 1, state_1;
- réaliser un décodeur qui permette l'affichage du code reçu (appui sur la touche A, affichage d'un caractère A).

Le module développé peut être utilisé comme un bloc participant au projet de montre digitale complet.

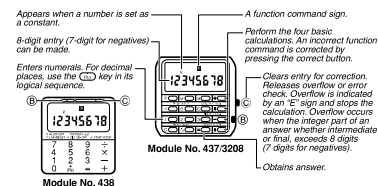
Operation Guide 437/438/3208 CASIO

Reading The Display



(Reversion to regular time) The watch reverts to the regular timekeeping display after operating in any mode.
(Auto return feature) The watch will return to the regular timekeeping display automatically if you do not perform any operation for five or six minutes in the calculator or alarm display.

Calculator Operation

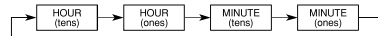


EXAMPLE	OPERATION	READ-OUT	EXAMPLE	OPERATION	READ-OUT
Basic calculation: $(12 - 0.5) \times 3 + 7$ $= 4.9285714 \dots$	12 \square \square 5 \square \square 3 \square + \square 7 \square =	4.9285714	$3 \times 4 = 12$ (4 is constant)	4 \square 0 \square 3 \square 0 \square =	12
Constant calculation: $3 + 4 = 7$ (4 is constant)	4 \square 0 \square 3 \square 0 \square =	7	$8 \times 4 = 32$	8 \square 0 \square 0 \square 3 \square 0 \square =	32
$8 + 4 = 12$	8 \square 0 \square 0 \square 3 \square 0 \square =	12	$3 + 4 = 0.75$	4 \square 0 \square 0 \square 3 \square 0 \square =	0.75
$3 - 4 = -1$	3 \square 0 \square 0 \square 3 \square 0 \square =	-1	$8 \div 4 = 2$	8 \square 0 \square 0 \square 3 \square 0 \square =	2
$8 - 4 = 4$	8 \square 0 \square 0 \square 3 \square 0 \square =	4	Power calculation: 2^3	3 \square 0 \square 0 \square 0 \square 3 \square 0 \square =	8
			3^2	3 \square 0 \square 0 \square 0 \square 3 \square 0 \square =	9

To save battery power, press the (B) button in the calculator mode to silence the tone. To retrieve sound, press the (B) button again.

Setting Daily Alarm Time

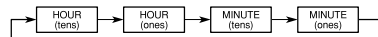
Every time the (C) button is pressed in the alarm time mode, the alarm-ON-mark (■) appears or disappears. When the alarm-ON-mark is lit, the buzzer sounds for 20 seconds at the preset time every day until cleared. To stop the buzzer while sounding, press any button.
(Setting the time signal) Every time the (C) button is pressed in the alarm time mode, the time-signal-ON-mark (A) appears or disappears. When the time-signal-ON-mark is lit, the watch sounds every hour on the hour.
(Sound demonstration) While both (C) and (C) buttons are pressed simultaneously, the buzzer sounds.
(Presetting alarm time)
1) Press (B) to set new alarm time.
2) Press (C) to set PM or AM.*
3) A numeral can be entered in the flashing position by pressing the required numeral key. The flashing digit can be shifted in the following order:



4) Press (B) to complete setting.
* When the watch is in the 24-hour system, the alarm time is displayed in that system.

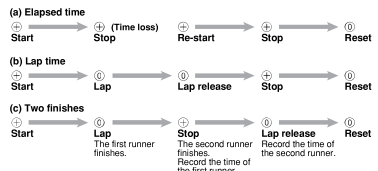
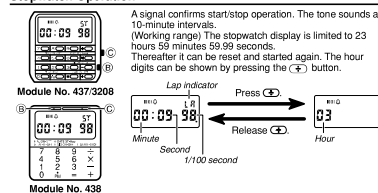
Setting Dual Time

1) Press (B) to set new time in dual time mode.
2) Press (C) to set PM or AM.*
3) A numeral can be entered in the flashing position by pressing the required numeral key. The flashing digit can be shifted in the following order:



4) Press (B) to complete setting.
* When the watch is in the 24-hour system, the dual time is displayed in that system.

Stopwatch Operation



Setting Time And Calendar

1) To set time and calendar, press (B) in the regular timekeeping mode.
2) Press the (C) key on a time signal to correct seconds.
3) Press (C) to move the flashing in the sequence shown below to select the other settings.



* AM or PM can be entered by pressing the (C) button.*
4) A numeral can be entered in the flashing position by pressing the required numeral key.
5) Press (B) to complete setting.
* While setting time and calendar, with every push of (C), the display is switched between 12-hour and 24-hour formats.

Operation Guide 437/438/3208 CASIO

Specifications

Accuracy at normal temperature: ± 15 seconds a month

Display capacity:

- Regular timekeeping
Hour, minute, second, am/pm, year, month, date and day on its double display
Time format: 12-hour and 24-hour
Calendar system:
• (Module No. 437/438) Auto-calendar pre-programmed to the year 2079
• (Module No. 3208) Auto-calendar pre-programmed from the year 2000 to 2099

- Calculator mode
8 digits (7 digits for negatives)
Abilities: Four basic calculations, chain & mixed operations, constants for $+/-$ / \times/ \div , power calculation
Decimal point: Full floating with underflow
Overflow check: Indicated by the 'E' sign, locking the calculator mode
- Stopwatch mode
Measuring unit: 1/100 second
Measuring capacity: 23 hours, 59 minutes, 59.99 seconds
Measuring modes: Elapsed time, lap time, two finishes
- Dual time mode
- Daily alarm mode
- Hourly time signal

Battery:

- (Module No. 437/3208)
• One lithium battery (Type: CR2016)
Approx. 5 years operation on type CR2016
(includes the alarm for 20 seconds and calculation for 1 hour per day)
- (Module No. 438)
• One lithium battery (Type: CR1616)
Approx. 3 years operation on type CR1616
(includes the alarm for 20 seconds and calculation for 1 hour per day)

NOTE: THERE IS NO WAY unit components can be damaged or malfunction, due to misoperation of buttons. If confusing information appears on the display it means entry sequence was incorrect. Please read the manual and try again.