

Sujet d'épreuves de la 48^e Compétition Nationale des Métiers

MÉTIER N°16 ELECTRONIQUE

MODULE C2

PROGRAMMATION EMBARQUEE – ALGORITHMIE

Soumis par :

Louis LEFEBVRE, Expert National WorldSkills France

Dominique CHATEAU, Expert Adjoint WorldSkills France

Référence du sujet : WSFR48CNAT-16-C2

Révision du sujet : 01

Date de diffusion : C2 - 17/10/2025

EXPLICATION DU MODULE

DUREE TOTALE DE L'ÉPREUVE	4 heures
DIFFUSION DU SUJET	Découvert le jour de la compétition : C2 - 17/10/2025

DESCRIPTION DU PROJET

Qu'est-ce que le temps ? Au-delà de l'une des grandeurs dont vous serez sûrement à court au cours des épreuves, le temps est une notion difficilement définissable, sujette aux plus grands débats des communautés scientifique et philosophique depuis des milliers d'années. Ce temps rythme pourtant le quotidien de toute l'humanité, et les avancées technologiques se targuent de toujours le mesurer de façon plus fine. Depuis l'avènement des technologies numériques, la maîtrise précise de l'écoulement du temps est même devenue un enjeu stratégique, tant pour le bon fonctionnement des machines (cadencement et synchronisation) que pour la souveraineté des Hommes (géolocalisation et navigation, météorologie, précision atomique pour la recherche et l'industrie...).

L'importance de la mesure de l'écoulement du temps ne date évidemment pas de l'apparition des technologies numériques : on en retrouve notamment le besoin depuis des siècles dans l'agriculture ou les rites religieux. Ce qui a significativement changé, en revanche, est la façon de se placer dans le temps et de dater les événements : parfois fonctions de la hauteur du soleil dans le ciel, parfois des cycles de la lune, parfois seulement de la volonté politique ou, aujourd'hui, calculés très précisément grâce aux fréquences des rayonnements électromagnétiques des électrons transitant entre les niveaux d'énergie des atomes de césium 133 par rapport à un événement religieux de référence survenu il y a plus de deux mille ans dont la date est incertaine, divers calendriers et conventions se sont succédé au fil des civilisations. Si l'humain usuel a seulement besoin de regarder sa montre et son calendrier au quotidien, nous vous proposons par ce sujet de découvrir certaines propriétés invraisemblables des conventions de temps et d'entrevoir la difficulté pour les historiens de dater correctement les événements du passé au regard des évolutions de notations.

Ainsi, le projet que vous réaliserez tout au long de la compétition est une carte électronique de calcul et de transmission pour un serveur de temps, appelée « *SyncOrSink* ». La carte que vous développerez aura pour fonction de mesurer le temps et de transmettre à un utilisateur ou à un équipement distant un horodatage, c'est-à-dire la date et l'heure à un instant précis.

Note : tout au long du document, le terme <REGION> est à remplacer par le trigramme associé à la région du compétiteur. Exemple : pour le compétiteur Auvergne Rhône-Alpes, le nom « 48_CNAT_16_Electronique_<REGION> » doit être remplacé par « 48_CNAT_16_Electronique_ARA ».

Région	Trigramme
Auvergne Rhône-Alpes	ARA
Bretagne	BRE
Hauts-de-France	HDF
Normandie	NOR
Nouvelle-Aquitaine	NAQ
Occitanie	OCC
Sud – Provence-Alpes-Côte-d'Azur	SUD
Sud – Provence-Alpes-Côte-d'Azur Parcours +	WPL

DESCRIPTION DU MODULE

Dans le cadre du module C2 portant sur l'évaluation des compétences en réflexion logique et programmation d'algorithmes, il est demandé au compétiteur de mesurer la charge CPU du microcontrôleur et d'implémenter un algorithme de calcul de la date et de l'heure.

Pour cela, il est, entre autres, mis à disposition du compétiteur :

- Une carte *SyncOrSink*, révision 1.8
- La spécification technique de la carte *SyncOrSink*
- Le schéma électronique de la carte *SyncOrSink*
- Le manuel de la HAL STM32
- Un projet STM32CubeIDE de départ implémentant une partie des fonctionnalités nécessaires pour l'épreuve

Livrables attendus

- Projet STM32CubeIDE « 48_CNAT_16_Electronique_<REGION>-C2 » complété avec les fonctionnalités attendues par le sujet.

Note

Seul le fonctionnement des fonctionnalités demandées sera évalué. La notation s'appuiera sur des tests en conditions réelles (manipulations directement sur la carte programmée) ainsi que sur des tests unitaires automatiques de la configuration du microcontrôleur. Les choix techniques, l'organisation et la qualité du code ne seront pas regardés par le jury. Le compétiteur est libre de modifier le projet fourni à sa guise.

Note : au sein de ce document, les composants physiques et les flux de données de la carte *SyncOrSink* présents sur le schéma électronique de la carte seront identifiés grâce à *ce style de mise en forme*. Les fonctions, constantes et variables logicielles du projet seront identifiées grâce à *ce style de mise en forme*.

TÂCHE 1 – CHARGE CPU

Dans cette tâche, il est demandé au compétiteur d'implémenter une fonction de calcul du taux d'utilisation du processeur (« charge CPU »). Soient `G_TIME_ELAPSED_SINCE_STARTUP` un compteur enregistrant le nombre de millisecondes écoulées depuis le démarrage du programme, et la constante `C_OPECYCLE_ALLOCATED_DURATION` = 5 ms la durée maximale d'exécution allouée à la boucle principale du programme. La charge CPU est définie comme le taux d'utilisation de la durée maximale allouée à l'exécution de la boucle principale du programme.

Il est demandé au compétiteur d'implémenter le calcul de la charge CPU de la boucle principale du programme et de l'afficher sur la page « état CPU » de l'écran utilisateur, au format décrit ci-dessous (en remplaçant « `xxx.xx` » par la charge CPU calculée pour l'itération précédente de la boucle principale).

Ligne	Caractères																
1	C	P	U		s	t	a	t	e								
2																	
3	C	P	U		l	o	a	d	:		x	x	x	.	x	x	%

Note

`G_TIME_ELAPSED_SINCE_STARTUP` étant un entier non signé 64 bits dont la durée de *rollover*, pour un incrément d'une microseconde, est significativement supérieure à la durée de cette épreuve, la prise en charge de son *rollover* n'est pas demandée.

Livrables attendus

- Implémentation de la fonction de calcul de la charge CPU.
- Affichage du pourcentage d'utilisation CPU sur la page « état CPU » de l'écran utilisateur.
- La charge CPU totale de ce module doit être inférieure à 60%.

INTRODUCTION AU CALCUL CALENDRAIRE

Soit le modèle UNIX un modèle de temps technique comptant le nombre de secondes écoulées depuis le 1^{er} janvier 1970 à 0h00 GMT+0 UTC (on appelle cet instant l'« Epoch UNIX »). Dans le modèle UNIX, chaque jour est considéré comme comptant exactement 86400 secondes (24 heures exactes).

Soit {TIMESTAMP} un horodatage UNIX quelconque, c'est-à-dire le nombre de secondes écoulées depuis l'Epoch, le 1^{er} janvier 1970 à 0:00:00 UTC.

Soient trois calendriers pris en charge par le système *SyncOrSink* : les calendriers julien, grégorien et UTC.

Pour la suite de ce sujet, il est attendu que le compétiteur implémente le calcul de la date et de l'heure dans chacun de ces trois calendriers à partir du {TIMESTAMP} UNIX fourni. Ces trois calendriers étant tous apparus avant l'introduction du modèle UNIX et à des époques différentes, leur date de début n'est pas la même ; afin de simplifier les calculs, la date de référence pour chaque calendrier (date à partir de laquelle les secondes sont comptées) sera la date de l'Epoch équivalente dans ledit calendrier :

Calendrier	Jour	Mois	Année	Heure
UNIX	1	Janvier	1970	0:00:00
Julien	19	Décembre	1969	0:00:00
Grégorien	1	Janvier	1970	0:00:00
UTC	1	Janvier	1970	0:00:00

L'implémentation des trois algorithmes devra être effectuée au sein de la même fonction *TIME_ComputeDate()*, prenant les paramètres suivants :

- Le calendrier à utiliser (énuméré)
- Le *timestamp* représentant l'instant dont la date et l'heure sont à calculer (entier non signé)

Et retournant une structure contenant les informations de date et d'heure.

Le compétiteur est autorisé à créer des sous-fonctions appelées par *TIME_ComputeDate()*.



La vérification de la validité de l'implémentation sera effectuée par tests unitaires automatiques de la fonction *TIME_ComputeDate()*. Ne pas modifier le prototype de la fonction, au risque que celle-ci ne puisse pas être appelée correctement par les outils de vérification.

Un ensemble de données de test est fourni en annexe 29 de ce sujet.

Afin d'aider le compétiteur dans sa tâche, l'outil « *TimestampToCalendar.exe* » est mis à la disposition du compétiteur. Cet outil, exécutable en ligne de commandes, permet d'obtenir le résultat attendu pour un *timestamp* UNIX donné. Voir le manuel d'utilisation annexe 37.

De plus, la date et l'heure calculées sont automatiquement affichées sur la page « horodatage » de l'écran utilisateur (programmation déjà effectuée, rien à faire par le compétiteur). Lorsque la page « horodatage » est affichée, l'appui sur le bouton *BTN_MODE* permet de modifier le calendrier affiché (programmation déjà effectuée, rien à faire par le compétiteur).

TÂCHE 2 – IMPLEMENTATION DU CALENDRIER JULIEN

Le calendrier julien, introduit par Jules César en 46 av. J.C. et utilisé jusqu'à son remplacement en 1582 par le calendrier grégorien (mais utilisé par certaines nations orthodoxes comme la Grèce jusqu'au XX^e siècle), implémente les calculs de temps de la façon suivante :

- Les années standard comptent 365 jours, avec 28 jours au mois de février
- Les années bissextiles comptent 366 jours, avec 29 jours au mois de février
- Est année bissextile toute année multiple de 4

Mois	Jan.	Fev.	Mars	Avr.	Mai	Juin	Jui.	Août	Sep.	Oct.	Nov.	Dec.
Nombre de jours année standard	31	28	31	30	31	30	31	31	30	31	30	31
Nombre de jours année bissextile	31	29	31	30	31	30	31	31	30	31	30	31

Ce calendrier ayant montré une forte dérive temporelle depuis son introduction (d'où sa réforme grégorienne), la date de référence équivalente à l'Epoch UNIX dans le calendrier julien n'est pas le 1^{er} janvier 1970, mais le 19 décembre 1969.

Dans cette tâche, il est attendu que le compétiteur implémente un algorithme de calcul de la date dans le calendrier julien, au sein de la fonction `TIME_ComputeDate()`.

Livrables attendus

- Implémentation d'un algorithme de calcul de l'horodatage dans le calendrier julien.

TÂCHE 3 – IMPLEMENTATION DU CALENDRIER GREGORIEN

Le calendrier grégorien, introduit par décret du pape Grégoire XIII le 14 février 1582 pour améliorer la correction des déviations temporelles du calendrier julien, conserve le format du calendrier julien, mais remplace la manière de calculer les années bissextiles. Tandis que le calendrier julien considérait bissextile toute année multiple de 4, l'ajustement au calendrier grégorien définit une année comme bissextile si elle respecte au moins l'un des deux critères suivants :

- L'année est divisible par 4 sans être divisible par 100
- L'année est divisible par 400

Ainsi, pour être bissextile, une année doit dans tous les cas être divisible par 4, mais elle doit aussi être divisible par 400 si elle est année de centenaire. Exemple : 2000, 2020, 2024, 2028 et 2400 sont bissextiles, mais pas 1900, 2100, 2200 ou 2300.

Dans cette tâche, il est attendu que le compétiteur adapte l'implémentation du calendrier julien de la tâche précédente pour y ajouter les paramètres du calendrier grégorien.

Livrables attendus

- Implémentation d'un algorithme de calcul de l'horodatage dans le calendrier grégorien.

TÂCHE 4 – IMPLEMENTATION DU CALENDRIER UTC

D'après Wikipédia :

UTC est une échelle de temps comprise entre le temps atomique international (TAI), qui est stable mais déconnecté de la rotation de la Terre, et le temps universel (TU) [temps astronomique], directement lié à la rotation de la Terre et donc lentement variable.

Dans « Temps Universel Coordonné » UTC, le terme « coordonné » indique que ce temps s'écoule de façon identique à celui du Temps Atomique International, il en a la stabilité et l'exactitude à un nombre entier de secondes près, et grâce à un mécanisme d'ajustement mis en œuvre chaque fois que nécessaire, il est décalé d'une seconde pour qu'il reste proche du temps universel à moins de 0,9 s près.

https://fr.wikipedia.org/wiki/Temps_universel_coordonné

Ainsi, le calendrier UTC est équivalent au calendrier grégorien, mais « chaque fois que nécessaire », une seconde artificielle (nommée « seconde intercalaire ») est manuellement ajoutée sur décision arbitraire du Bureau International des Poids et Mesures. Ces ajouts ou retraits peuvent être réalisés soit le 31 décembre à minuit, soit le 30 juin à minuit^[1]. La relation entre les temps UNIX et UTC n'étant de fait pas bijective, deux temps UTC peuvent avoir le même temps UNIX :

Etape	Temps UTC	Temps UNIX	
1	2008-12-31 23:59:57	1 230 767 997	
2	2008-12-31 23:59:58	1 230 767 998	
3	2008-12-31 23:59:59	1 230 767 999	
4	2008-12-31 23:59:60	1 230 768 000	Ajout d'une seconde intercalaire : 23:59:60
5	2009-01-01 00:00:00	1 230 768 000	
6	2009-01-01 00:00:01	1 230 768 001	
7	2009-01-01 00:00:02	1 230 768 002	



Lors de la 27^e Conférence générale des poids et mesures (18/11/2022, Versailles, France), face à la complexité croissante de maintenir un tel système, la communauté scientifique internationale a voté à la quasi-unanimité la suspension des secondes intercalaires à partir de 2035. Le temps UTC fonctionnera sans ces secondes intercalaires jusqu'en 2135, l'hypothèse étant qu'au cours des 100 prochaines années, les métrologues auront trouvé un meilleur système pour synchroniser les échelles de temps atomique et astronomique.

¹ Cette correction temporelle est décidée tous les 6 mois par le Bureau International des Poids et Mesures (BIPM) dans son Bulletin C. Le bulletin annonce soit un ajout de seconde intercalaire, soit une suppression, soit qu'il n'y aura aucun des deux pour la prochaine période de 6 mois.

Ci-dessous, la liste des secondes intercalaires ajoutées depuis leur introduction en 1972 :

Année	30 juin 23:59:60	31 déc. 23:59:60
1972	78796800	94694400
1973	0	126230400
1974	0	157766400
1975	0	189302400
1976	0	220924800
1977	0	252460800
1978	0	283996800
1979	0	315532800
1980	0	0
1981	362793600	0
1982	394329600	0
1983	425865600	0
1984	0	0
1985	489024000	0
1986	0	0
1987	0	567993600
1988	0	0
1989	0	631152000
1990	0	662688000
1991	0	0
1992	709948800	0
1993	741484800	0
1994	773020800	0
1995	0	820454400
1996	0	0
1997	867715200	0
1998	0	883612800

Année	30 juin 23:59:60	31 déc. 23:59:60
1999	0	0
2000	0	0
2001	0	0
2002	0	0
2003	0	0
2004	0	0
2005	0	1136073600
2006	0	0
2007	0	0
2008	0	1230768000
2009	0	0
2010	0	0
2011	0	0
2012	1341100800	0
2013	0	0
2014	0	0
2015	1435708800	0
2016	0	1483228800
2017	0	0
2018	0	0
2019	0	0
2020	0	0
2021	0	0
2022	0	0
2023	0	0
2024	0	0

Code couleur	Signification
0	Pas de seconde intercalaire
Timestamp*	Ajout d'une seconde intercalaire

* Le timestamp indiqué est le timestamp UNIX correspondant à la seconde intercalaire 23:59:60.

Dans cette tâche, il est attendu que le compétiteur adapte l'implémentation du calendrier grégorien de la tâche précédente pour y ajouter les paramètres du calendrier UTC. Les temps UNIX et UTC n'étant pas bijectifs (UTC est une surjection dans UNIX), il est possible d'établir un algorithme de conversion de UTC vers UNIX, mais théoriquement impossible de convertir UNIX vers UTC (puisqu'il peut y avoir deux temps UTC pour un même temps UNIX). Dans le cas des secondes intercalaires, il est attendu que le compétiteur donne le résultat pour la seconde intercalaire (23:59:60) à la place de la seconde ronde (0:00:00).

Livrables attendus

- Implémentation d'un algorithme de calcul de l'horodatage dans le calendrier UTC.

LISTE DES ANNEXES

Annexes applicables à ce module :

Identifiant	Nom
Documents de compétition	
01	Liste des annexes de compétition
02	Planning CNAT 48 - Electronique
03	Liste du matériel autorisé
04	Barème global – Electronique
08	Barème résumé – Electronique – Module C2
29	Données de test module C2
Manuels d'utilisation	
13	Abréviations et unités
18	STM32CubeIDE – Manuels d'utilisation
37	TimestampToCalendar – Manuel d'utilisation
Documentations techniques et normes	
28	Documentations technique STM32 (HAL + registres)
34	Documentation technique ADA1655
35	Documentation technique AT27C010
Documents de conception carte <i>SyncOrSink</i>	
36	Spécification technique de la carte
30	Schéma électrique complet de la carte
31	Plan d'implantation de la carte
32	Liste des composants de la carte (BOM)
33	Notice de programmation de la carte