

Synchronisation PTP de caméras RGB et événementielles

Journée Outils Logiciels et Matériels pour la Recherche sur les
Véhicules Terrestres Autonomes

Stéphane Bonnet

UMR CNRS 7253 Heudiasyc

5 octobre 2023

Synchronisation locale

- Besoin de synchroniser les données issues de capteurs d'un même véhicule
- Synchronisation *locale* à l'échelle d'un véhicule



Synchronisation globale

- Besoin de synchroniser les données issues de capteurs de véhicules différents
- Synchronisation *globale* à l'échelle d'un réseau de véhicules et / ou d'infrastructures



Objectif

Permettre de *corrél*er les données brutes ou traitées issues du système de véhicules:

- *Fusion multicapteurs distribuée ou non*
- *Redondance de perception*
- *Intégrité*
- *Collaboration*
- ...

Objectif de synchronisation

L'objectif de synchronisation est de *corriger* les horloges des capteurs afin de les rendre *cohérentes*:

- *Synchronisation de fréquence*: ajustement de la fréquence des horloges – le temps s'écoule à la même vitesse
- *Synchronisation de phase*: ajustement du décalage temporel entre horloges – les instants sont communs

Le temps peut être absolu ou relatif:

- *Temps absolu*: temps UTC, GNSS, ...
- *Temps relatif*: temps écoulé depuis un événement (mise en route du système, d'un capteur), ...

Quel temps choisir?

- À priori, localement le temps absolu n'est pas indispensable: le temps relatif suffit à condition que l'événement de départ soit le même pour tous les capteurs embarqués.
- En revanche, globalement, le temps absolu est indispensable pour pouvoir corréler les données issues de capteurs de véhicules différents.

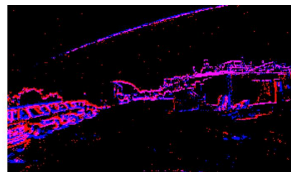
En pratique, il est nécessaire de disposer d'un temps absolu *local* pour pouvoir synchroniser les horloges des capteurs et corréler les données. En général on utilise le temps UTC fourni par un récepteur GNSS.

Moyens de synchronisation

- *Synchronisation logicielle*: utilisation d'un protocole de synchronisation logiciel (NTP, PTP, ...)
- *Synchronisation matérielle*: utilisation d'un protocole de synchronisation matérielle (Triggers matériels, PPS, ...)

Caméras événementielles Prophesee EVK4

- Chaque pixel est un capteur indépendant
- Seuls les pixels qui détectent un changement d'intensité lumineuse sont transmis
- Les événements $\{x, y, t, p\}$ sont transmis en lots
- Les événements sont datés par rapport à un compteur interne précis mais relatif au démarrage de la caméra
- Interface USB 3.0
- Entrée de synchronisation: signal physique



Caméras RGB Allied Vision Prosilica

- Caméras industrielles haute résolution
- Interface GigE
- Supportent le protocole PTP (Precision Time Protocol)
- Génèrent un signal de synchronisation en sortie (Trigger) sur ouverture du shutter virtuel



Problématique

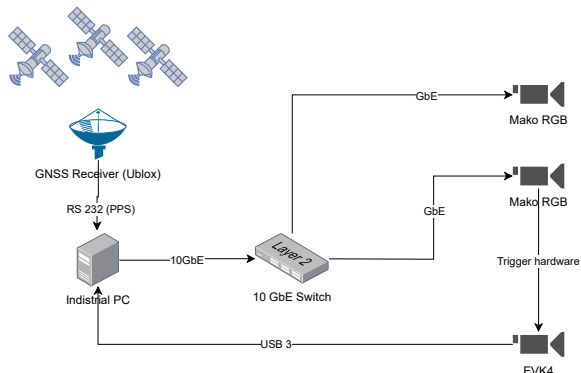
- Synchroniser les caméras RGB et événementielles pour corrélérer les données issues des deux types de caméras.
- Et aussi avec d'autres capteurs (LiDAR en particulier)
- Et aussi avec d'autres caméras événementielles sur d'autres véhicules. . .

Objectifs

- Synchronisation globale sur le temps UTC
- Exigence de précision: inférieure à 0.1 ms

Architecture matérielle

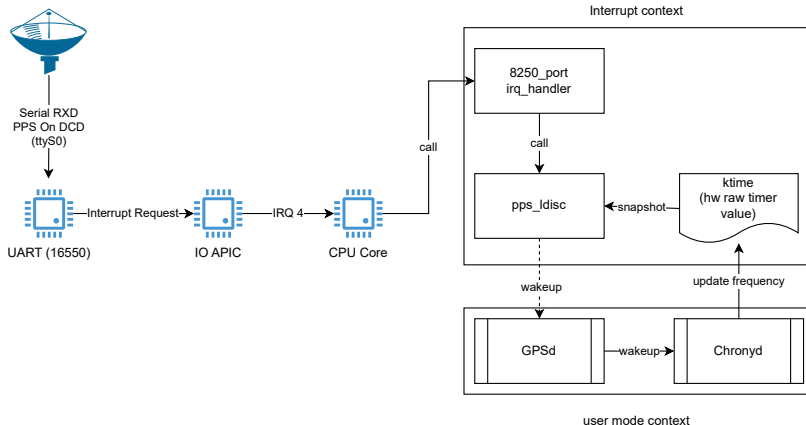
- PC Intel Core i7 / Linux
- Switch 10GbE POE
- Récepteur GNSS Ublox EVK-8T



Architecture logicielle

- Synchronisation globale sur le temps UTC de l'horloge interne du PC: NTP (*Network Time Protocol*)
 - Démon GPSd pour la récupération des données du récepteur GNSS: GNZDA, PPS
 - Démon Chrony pour la synchronisation de l'horloge interne (*Realtime clock*) du PC
- Synchronisation locale des caméras sur le temps UTC de l'horloge interne du PC: PTP
 - Démon phc2sys pour la synchronisation des horloges des interfaces Ethernet avec l'horloge interne du PC
 - Démon ptp4l pour la diffusion du temps UTC sur le réseau (rôle de Grand Master)

Synchronisation globale: NTP, GPSd, Chrony



Synchronisation globale: NTP, GPSd, Chrony

- Le noyau prend un snapshot de l'horloge temps-réel matérielle à l'instant du PPS
- Le démon GPSd récupère la trame ZDA (heure UTC réelle) et le snapshot PPS (heure UTC du PC à l'instant du PPS)
- Le démon Chrony utilise ces données pour synchroniser l'horloge interne du PC
 - Ajustement de la fréquence si l'erreur est faible (principe de la PLL)
 - Saut si l'erreur est trop importante

PTP: Precision Time Protocol (IEEE 1588)

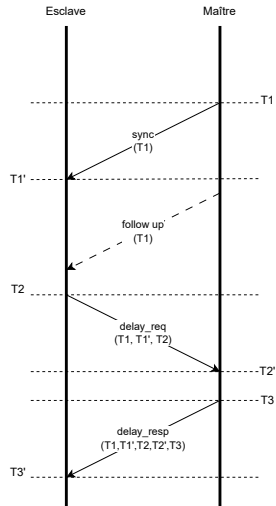
- Fonctionnement maître-esclave
- Exploite les capacités de timestamping des interfaces Ethernet
- Timestamping hardware des paquets à l'émission / réception
- But: obtenir un consensus global sur les horloges internes des équipements du réseau
- Précision de l'ordre de la microseconde

IEEE 802.1AS

- Profil simplifié de PTP pour les réseaux filaires et le *Time-Sensitive Networking*
- Profil retenu dans l'automobile pour la synchronisation des futures architectures *Software-defined vehicles* basées sur Ethernet

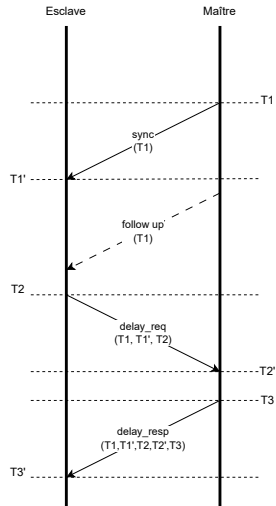
PTP: Principe 1/2

- But: déterminer l'offset $o = s - m$ entre l'horloge du maître m et l'horloge de l'esclave s malgré le délai d de transmission des paquets
- Sync: le maître envoie la date T_1 aux esclaves.
- Sync: l'esclave note la date T_1' de réception de Sync. $T_1' - T_1 = o + d$.



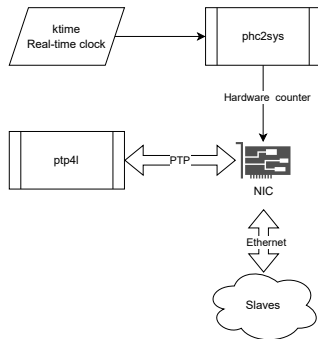
PTP: Principe 2/2

- Delay_req: l'esclave envoie la date T_2 au maître.
- Delay_req: le maître note la date T'_2 de réception de Delay_req. $T'_2 - T_2 = -o + d$.
- Delay_resp: les esclaves calculent $(T'_1 - T_1) - (T'_2 - T_2) = o + d - (o + d) = 2o$.
- L'offset est connu et ils ajustent leurs horloges.
- On suppose que le délai de transmission est symétrique.



Implémentation sous Linux

- phc2sys: synchronise l'horloge hardware de la carte réseau avec l'horloge du PC (asservie NTP)
- ptp4l: configuré en *Grand Master*, diffuse le temps sur le réseau



Sources possibles d'imprécision et d'incertitudes

- Précision du signal PPS du récepteur GNSS
- Latences liées aux interruptions matérielles et aux préemptions
- Latences de la mise à jour des timers hardware dans le noyau
- Effets de la synchronisation PTP par phc2sys, asservie sur l'horloge temps-réel NTP

Focalisation sur le second point: la latence d'interruption. C'est le temps indéterminé entre l'occurrence d'un événement physique et son traitement par le système d'exploitation.

Mesure de la latence d'interruption

- Instrumentation du noyau Linux pour affirmer un signal physique (signal RTS sur le port série) lors du traitement de l'interruption liée au PPS (IRQ 4 sur PC, pour le premier port série)
- Comparaison de ce signal avec le signal PPS du récepteur GNSS
- Comparaison de ce signal avec le trigger en sortie de la caméra RGB. La caméra RGB est configurée pour déclencher une acquisition à la seconde UTC exacte.

Ceci permet d'observer à l'oscilloscope les relations temporelles entre les différents signaux.

Que peut-on contrôler?

Les sources de latence d'interruption sont multiples:

- Occurrence pendant le traitement d'une autre interruption (interruptions non préemptibles dans les noyaux non PREEMPT_RT)
- Occurrence pendant le traitement d'une section critique noyau
- Traitement sur un cœur dans un état basse consommation qui doit être préalablement réactivé. . .

Seul le dernier point est maîtrisable dans le noyau Linux standard.

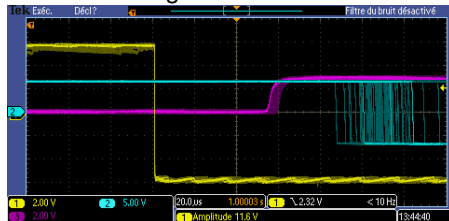
Contrôle des modes de gestion de l'alimentation (C-states) Intel

- Désactivables globalement par le paramètre de ligne de commande noyau
`intel_idle.max_cstate=0`
- Désactivables individuellement par la commande
`cpupower -c <cpu> idle-set -D 0`
- Dans ce cas, il faut affecter l'affinité de l'IRQ PPS au cœur qui reste actif:
`echo <cpu> > /proc/irq/<irq>/smp_affinity`

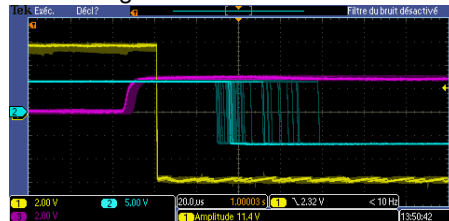
Nehalem					Mperf			Idle_Stats									
CPU	C3	C6	PC3	PC6	C0	Cx	Freq	POLL	C1	C1E	C3	C6	C7s	C8	C9	C10	
0	0,00	0,00	0,00	0,00	99,74	0,26	3600	97,33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
1	0,00	0,00	0,00	0,00	99,74	0,26	3599	97,49	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
2	0,15	7,85	0,00	0,00	1,51	98,49	3600	0,00	0,00	0,09	0,19	8,27	0,00	18,16	0,11	71,70	
3	0,05	4,27	0,00	0,00	1,32	98,68	3600	0,00	0,01	0,79	0,05	4,51	0,01	21,86	0,05	71,40	
4	0,10	3,72	0,00	0,00	2,83	97,17	3600	0,00	0,08	0,59	0,14	4,00	0,00	12,29	0,00	80,09	
5	0,09	2,46	0,00	0,00	2,09	97,91	3599	0,00	0,08	0,24	0,13	2,64	0,00	5,57	0,00	89,26	
6	1,30	10,29	0,00	0,00	17,15	82,85	3600	0,00	1,45	17,88	2,61	13,01	0,00	13,04	0,00	34,90	
7	0,11	3,85	0,00	0,00	1,89	98,11	3599	0,00	0,04	3,58	0,11	4,09	0,00	24,73	0,00	65,57	

Résultats sans désactivation des C-states

CPU non chargé



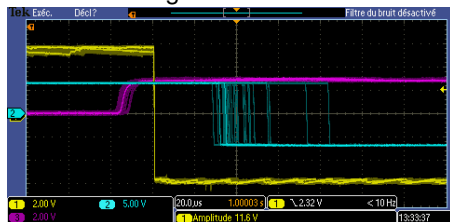
CPU chargé



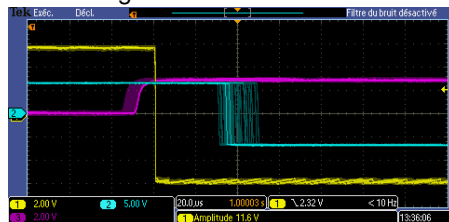
- Trace jaune: PPS
- Trace bleue: signal RTS dans le gestionnaire d'interruption
- Trace magenta: trigger de la caméra RGB

Résultats avec désactivation des C-states et affinité sur CPU physique 0

CPU non chargé



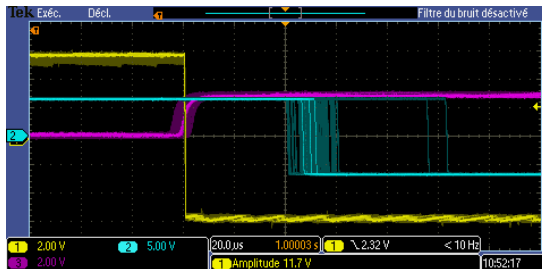
CPU chargé



- Trace jaune: PPS
- Trace bleue: signal RTS dans le gestionnaire d'interruption
- Trace magenta: trigger de la caméra RGB

Résultat après réglage de l'offset PPS dans Chrony

On obtient une précision de l'ordre de $\pm 6\mu\text{s}$.



- Trace jaune: PPS
- Trace bleue: signal RTS dans le gestionnaire d'interruption
- Trace magenta: trigger de la caméra RGB

Journée “Caméra à événements appliquée à la robotique...”

... organisée par le GDR ISIS à Paris Jussieu le 16 novembre 2023.

<https://www.gdr-isis.fr/index.php/reunion/499/>

Merci de votre attention!