Semaine du 13/10/2017

Valeurs des IMUs :

Configuration actuelle,

Accelerometre : sensibilité +- 2g, pour obtenir la valeur en G diviser par 16384.

Gyrometre : sensibilité +-250 ° /s, pour obtenir en °/s diviser par 131

Temperature : signé sur 16 bits, pour obtenir en °C, temperature/340+36.53

Magnetometre : +- 0.88 Gauss, pour obtenir valeur en Gauss divisé par 0.73

Pour créér plus de 2 taches, changer le configTOTAL\_HEAP\_SIZE à (8192) dans le fichier FreeRTOSConfig.h

Fonctionnement de la Queue sous FreeRTOS:

Création d’un Handle pour le Queue, global :

QueueHandle\_t xQueue;

Création d’une queue :

Queue\_Id = xQueueCreate(items\_number, size\_of\_items);

Passer la queue en argument aux taches qui manipulent la queue:

xTaskCreate(vQueueRead, "Q1", configMINIMAL\_STACK\_SIZE + 200, **(void\*) xQueue**, 3, &xQHandleRead );

Pour ecrire dans le Queue:

xQueueSendToBack(Queue\_id, (void\*) items\_to\_write, ( TickType\_t ) int);

Pour lire la queue:

xQueueReceive(Queue\_id, Buffer\_to\_receive , (TickType\_t) int);

Utilisation des notifications

Les notifications permettent de synchroniser les taches

Pour bloquer une tache en attente d’un signal venant d’une autre tache :

ulTaskNotifyTake( pdTRUE, portMAX\_DELAY );

Pour envoyer un signal à une tache (et donc debloquer cette taches) :

xTaskNotifyGive(TaskHandle);

Utilisation des semaphores/mutex

Créer un handle pour la semaphore :

static SemaphoreHandle\_t ma\_semaphore;

Créer la semaphore, ici un mutex:

ma\_semaphore = xSemaphoreCreateMutex();

Puis activer le mutex :

xSemaphoreTake(ma\_semaphore, portMAX\_DELAY);

et ne pas oublier de la lever après utilisation de la ressource :

xSemaphoreGive(ma\_semaphore);

Delay

Dans les taches utiliser les vTaskDelay() pour ne pas bloquer FreeRTOS et permettre a scheduler de basculer entre les taches.

Fonctionnement de l’application Android envisagé : Appui sur un bouton de la Nordic pour lancer l’acquisition des données IMU, et nouvel appui pour arreter la capture.

A afficher sur l’application : indicateur de capture du mouvement, et differences entre les deux bras (un bras sain et un bras malade).

Pour la SD card :

Attention instance utilisé -> si instance0 du spi utilisé on doit utilisé une instance différente pour le twi. Mais le changement du twi a l’instance 1 dans le sdk ne suffit pas il faut aussi l’indiqué dans le main avec la ligne static nrf\_drv\_twi\_t const m\_twi = NRF\_DRV\_TWI\_INSTANCE(0);

Les interruption freeRTOS n’interrompt donc pas l’écriture => pas besoin d’utiliser.