

Introduction au TP3

Francesco Mondada, Frank Bonnet
Eliot Ferragni, Daniel Burnier
IEM - STI - EPFL



Introduction

Il y a trois objectifs principaux pour ce travail pratique:

- **La prise en main de ChibiOS**, le système RTOS que vous allez utiliser lors des trois prochains TPs ainsi que le mini-projet.
- **Compréhension des Threads** et des différentes méthodes de gestion de la fréquence et des priorités des Threads.
- **La gestion d'un capteur, l'IMU**, pour effectuer des mesures de l'accélération et de vitesse de rotation du robot en temps réel.

ChibiOS

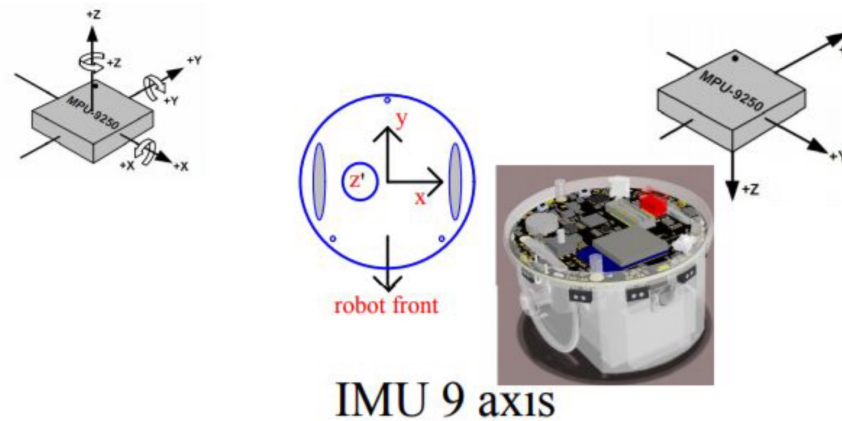
Comme vu lors du cours 4, dans un RTOS comme ChibiOS, l'implémentation des fonctionnalités du code sont implémentées dans des **Threads (taches)**, qui vont être séquencés par le Scheduler du RTOS pour être exécutés avec des timing déterminés par le programmeur.

Le code que vous allez prendre en main contient déjà une bonne partie de l'implémentation des Threads que vous allez utiliser durant ce TP. **Votre tâche sera tout d'abord de jouer avec les paramètres de configuration, puis d'implémenter le code de fonctionnalité à l'intérieur des Threads.**

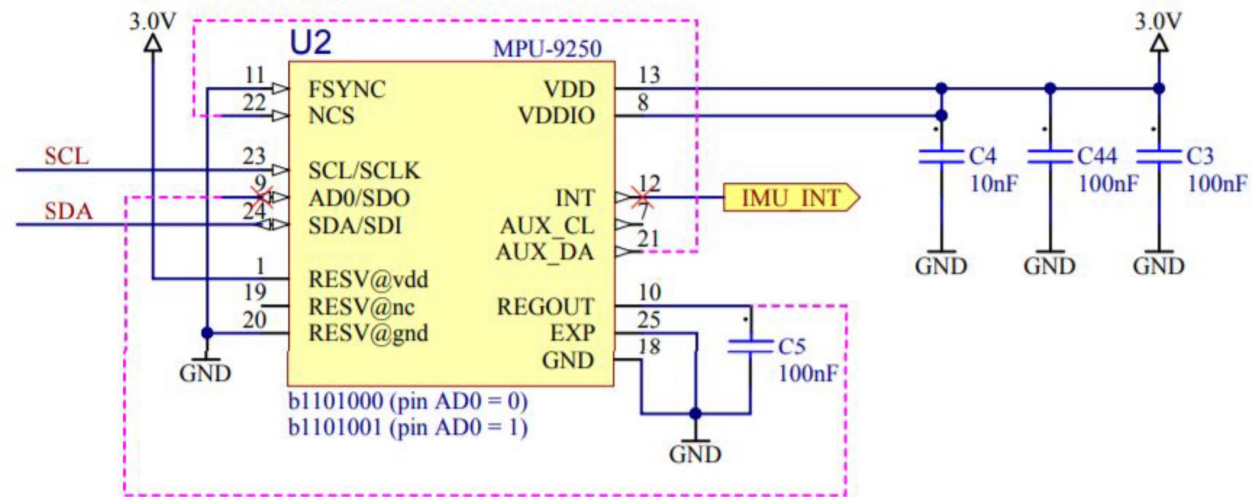
Prise en main de ChibiOS

1. Lisez bien les consignes, il y a de la théorie contenue dans la consigne qui doit être comprise en parallèle de ce qui a été vu en cours.
2. On vous conseille dans un premier temps d'utiliser les Threads déjà implémentés pour vous plutôt que d'en réimplémenter vous-même.
3. Explorez le code, en parcourant les différents fichiers de définitions des différentes librairies.
4. Ensuite vous pouvez implementer vos threads

IMU MPU-9250



IMU 9 axis



IMU MPU-9250

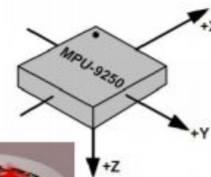
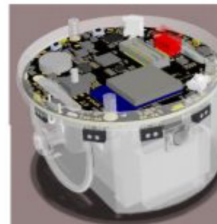
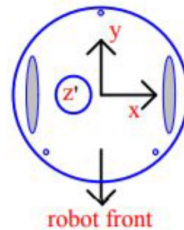
1. Accéléromètre (3 axes)
2. Gyroscope (3 axes)
3. Magnétomètre (3 axes) (inutilisable sur la majeure partie des e-puck2 à cause de problèmes de PCB)

IMU MPU-9250

Suggestion pour le TP:

- Observez dans le dossier électronique où est positionné physiquement l'accéléromètre sur le PCB du robot. Pourquoi cette position ?
- Trouvez le datasheet de l'IMU sur le net et observez les valeurs physiques qu'il est capable de mesurer (intensité des accélérations, etc.)

The diagram shows a 3D perspective of the MPU-9250 sensor. It is a rectangular block with 'MPU-9250' printed on its top surface. A vertical arrow labeled '+Z' points upwards from the center of the top surface. A curved arrow around the +Z axis indicates rotation. Two horizontal axes are shown: one pointing to the right labeled '+Y' and one pointing towards the viewer labeled '+X'. Curved arrows around the +Y and +X axes indicate rotation. The sensor is mounted on a base, with two lines representing connection points or wires extending from the bottom left.



STM32F4

Interface I2C

3.0V

U2 MPU-9250

FSYNC 11
NCS 22

SCL 23
SCL/SCLK 23
AD0/SDO 9
SDA/SDI 24

RESV@vdd 19
RESV@nc 20
RESV@gnd 21

b1101000 (pin AD0 = 0)
b1101001 (pin AD0 = 1)

VDD 13
VDDIO 8

INT 12
AUX_CL 21
AUX_DA 21

REGOUT 10
EXP 25
GND 18

IMU INT

C4 10nF
C44 100nF
C3 100nF

3.0V

GND

C5 100nF

ChibiOS / IMU

Couches d'abstraction:

```
int main(void)
{
    /* System init */
    halInit();
    chSysInit();
    serial_start();
    timer11_start();
    i2c_start();
    imu_start();
}
```

main.c

Récupération des valeurs de l'IMU et calcul du vecteur Accélération.

```
typedef struct {
    float acceleration[3]; // m/s^2
    float gyro_rate[3]; // rad/s
    float temperature;
    float magnetometer[3]; // uT
    int16_t acc_raw[3];
    int16_t gyro_raw[3];
    int16_t acc_offset[3];
    int16_t gyro_offset[3];
    int16_t acc_filtered[3];
    int16_t gyro_filtered[3];
    uint8_t status;
} imu_msg_t;
```

imu.c/imu.h

Capture des données de l'IMU (Thread) et partage de ces données (MessageBus).

```
void mpu9250_change_addr(void) {
    if(imu_addr == MPU9250_ADDRESS_AD1_0) {
        imu_addr = MPU9250_ADDRESS_AD1_1;
    } else {
        imu_addr = MPU9250_ADDRESS_AD1_0;
    }
}
```

mpu9250.c/
mpu9250.h

Librairie de gestion du device IMU (registres de configuration, etc.)

```
/** Starts the I2C interface */
void i2c_start(void);

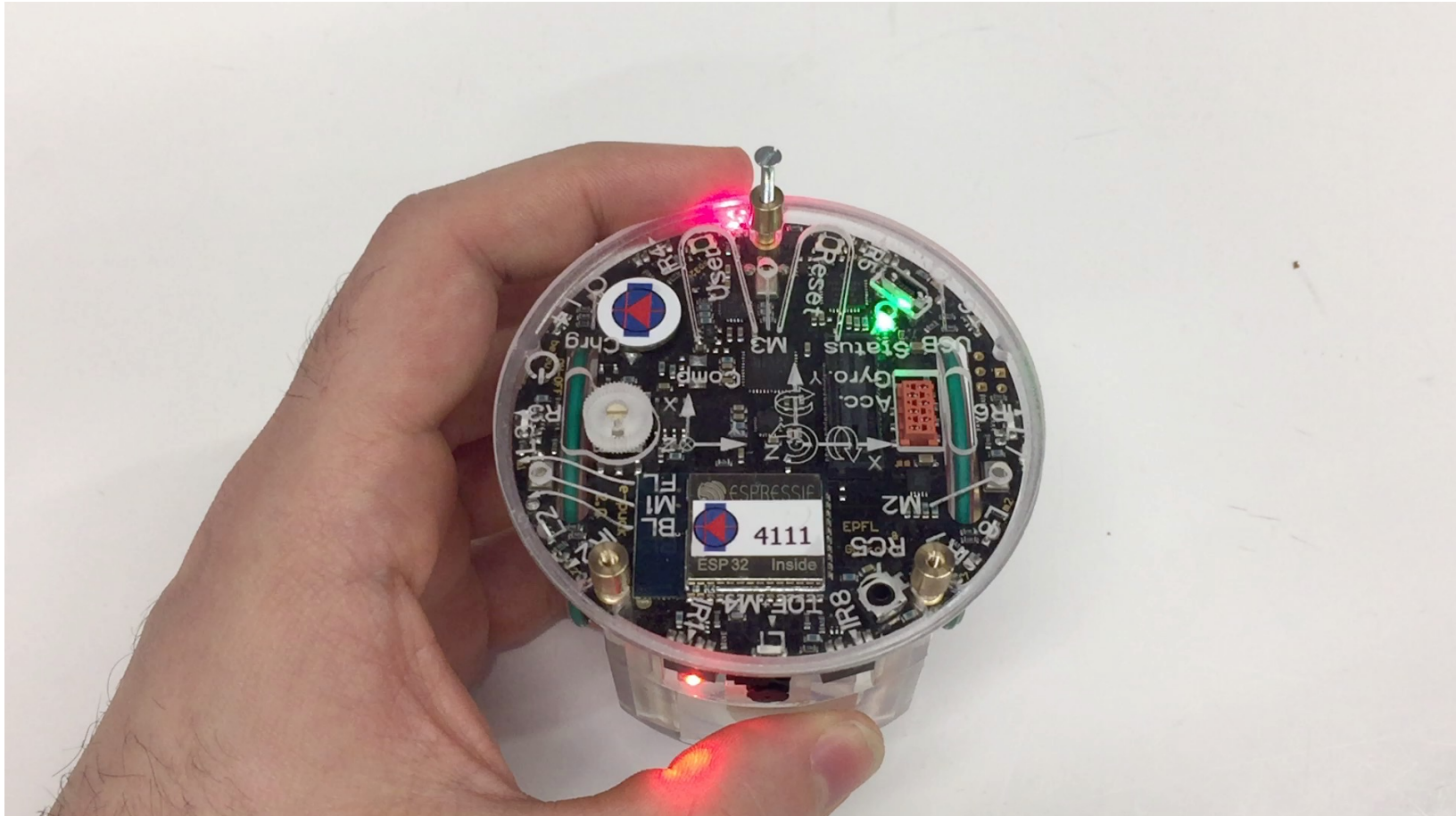
/** Stops the I2C module */
void i2c_stop(void);

/** Gets the last I2C error */
```

i2c_bus.c/i2c_bus.h

librairie de gestion de l'I2C pour communiquer entre le STM32F4 et le MPU 9250

Démo



Enjoy TP3...