

Informe de Avances

Moises Stevend Meza Rodriguez

Semana 1.-

- Se optimizó el algoritmo de control de trayectorias usando posiciones relativas basados en el algoritmos de posiciones del los smartphones implementados en C++.
- Se corrigió las vías del circuito electrónico para que el hardware pueda tener un control de carga de batería optimizado.

Semana 2.-

- Se continuó con los algoritmos de trayectorias, y se agrego las funciones para capturar las 10 trayectorias asignadas al inicio.
- Se implementó una API en el microcontrolador ESP32 para la comunicación mediante Bluetooth con el smartphone (android).

Semana 3.-

- Se optimizó la API adicionando paquetes JSON (tramas de comunicación) con la versión actual del APP.
- Se tuvo una reunión con el personal encargado del desarrollo del APP del Health Recover y se procedió a entregar una versión actualizada del hardware para su evaluación de la comunicación con el APP.

Semana 4.-

- Se optimizó el algoritmos usando los dos núcleos del ESP32.
- Se coordinó para actualizar la versión actual modular a una versión superficial.

Observaciones.-

Se pudo probar el seguimiento de trayectorias con una buena fidelidad, pero para movimientos no tan cercanos a la trayectoria original pierde la secuencia, para ello es necesario hacer más pruebas y/o calibrar con más movimientos.

Cabe resaltar que durante el mes de enero se trabajó en el código mayormente, salvo las correcciones para la carga de batería.

Los códigos se subieron al github del proyecto (repositorio privado):

https://github.com/healthRecover/proyecto_lid

```
Atom
proyecto_HR2.ino — proyecto_HR2 — ~/Arduino/proyecto_HR2 — Atom

Project
proyecto_HR2
proyecto_H

proyecto_HR2.ino
83 //*****External interruption*****//
84 void IRAM_ATTR isr(){
85   //yield();
86   delay(10);
87   if(fpress){
88     led_plot_Battery(gpio_leds);
89   }else{
90     clear_led(gpio_leds);
91   }
92   fpress = not(fpress);
93 }
94 //*****//
95
96
131 // Now set up two tasks to run independently.
132 xTaskCreatePinnedToCore(
133   TaskBluetooth
134   , "TaskBluetooth" // A name just for humans
135   , 100000 // This stack size can be checked & adjusted by
136   , NULL
137   , 1 // Priority, with 3 (configMAX_PRIORITIES - 1) being
138   , &Task1
139   , 0);
140
141 xTaskCreatePinnedToCore(
142   TaskGesture
143   , "TaskGesture"
144   , 100000 // Stack size
145   , NULL
146   , // ...

proyecto_HR2.ino
152 void loop()
153 {
154   if(enabled==true){
155     switch(move){
156       case 1:
157         if(set_en_m2==false){
158           imu.begin();
159           imu.setSensors(INV_XYZ_GYRO); // Enable gyroscope only
160           imu.setGyroFSR(2000); // Set gyro to 2000 dps
161
162           imu.dmpBegin(DMP_FEATURE_GYRO_CAL | // Enable gyro cal
163             DMP_FEATURE_SEND_CAL_GYRO, // Send cal'd gyro values
164             100);
165           set_en_m1=false;
166           set_en_m2=true;
167         }
168         move1(); // Set DMP rate to 10 Hz
169         break;
170       case 2:
171         if(set_en_m1==false){
172           imu.begin();
173           imu.setSensors(INV_XYZ_GYRO | INV_XYZ_ACCEL | INV_XYZ_COMPASS)
174           imu.dmpBegin(DMP_FEATURE_ANDROID_ORIENT);
175           imu.dmpSetOrientation(orientationMatrix);
176
177           set_en_m1=true;
178           set_en_m2=false;
179         }
180
181         move2();//listo
182         break;
```

Fig 1.- Vista general del código en diferentes pantallas