Location sub-system IO

2 functions

* getCoordinates();
* getHeading();

These 2 functions purpose are self explanatory

3 global variables will be updated by the functions

* int x\_loc – is the mm distance along x-axis from the (0,0) coordinate of the grid which is set in parameters this will be an integer value between 0mm and 5000mm
* int y\_loc – is the mm distance along y-axis blah blah
* int heading – is the magnetic heading in degrees. This is an integer value between 0 and 360 degrees

Ideally these functions will be called once during each cycle of the main loop. With pozyx preamble length set to 4096 (this offers the best accuracy) and number of readings to average set to 5(arbitrary choice that can be adjusted) the 2 functions together take approx. 540ms to run. This offers a steady heading which didn’t drift when observing and a definite accuracy of <100mm (it was actually more like <40mm but I didn’t want to check through thousands of readings).

Code follows:

#include <Pozyx.h>

#include <Pozyx\_definitions.h>

#include <Wire.h>

////////////////////////////////////////////////

////////////////// PARAMETERS //////////////////

////////////////////////////////////////////////

const int num\_to\_avg = 5; //number of position readings to average out for the output of position data

const uint8\_t num\_anchors = 4; // the number of anchors

uint16\_t anchors[num\_anchors] = {0x6827, 0x6821, 0x687c, 0x6851}; // the network id of the anchors

int32\_t anchors\_x[num\_anchors] = {0, 4900, 0, 4900}; // anchor x-coorindates in mm

int32\_t anchors\_y[num\_anchors] = {0, 0, 3900, 3900}; // anchor y-coordinates in mm

int32\_t heights[num\_anchors] = {750, 1700, 1700, 750}; // anchor z-coordinates in mm

uint8\_t algorithm = POZYX\_POS\_ALG\_UWB\_ONLY; // positioning algorithm to use. try POZYX\_POS\_ALG\_TRACKING for fast moving objects.

uint8\_t dimension = POZYX\_2D; // positioning dimension ie 2D, 2\_5D and 3D

int32\_t height = 0; // The z position of pozyx in mm

int x\_loc = -99; //The measured x location in mm from coordinate (0,0)

int y\_loc = -99; //The measured y location in mm from coordinate (0,0)

int heading = -99; //The magnetic heading in degrees

////////////////////////////////////////////////

// function to manually set the anchor coordinates

// supplied by pozyx website

void setAnchorsManual(){

for(int i = 0; i < num\_anchors; i++){

device\_coordinates\_t anchor;

anchor.network\_id = anchors[i];

anchor.flag = 0x1;

anchor.pos.x = anchors\_x[i];

anchor.pos.y = anchors\_y[i];

anchor.pos.z = heights[i];

Pozyx.addDevice(anchor, NULL);

}

if (num\_anchors > 4){

Pozyx.setSelectionOfAnchors(POZYX\_ANCHOR\_SEL\_AUTO, num\_anchors, NULL);

}

}

void setup() {

// put your setup code here, to run once:

Serial.begin(115200);

if(Pozyx.begin() == POZYX\_FAILURE){

Serial.println(F("ERROR: Unable to connect to POZYX shield"));

Serial.println(F("Reset required"));

delay(100);

abort();

}

// clear all previous devices in the device list

Pozyx.clearDevices(NULL);

//calibration

setAnchorsManual();

// sets the positioning algorithm

Pozyx.setPositionAlgorithm(algorithm, dimension, NULL);

delay(1000); //delay not required

}

// get position information

void getCoordinates (){

coordinates\_t position;

int status;

int out\_x;

int out\_y;

for (int i = 0; i<num\_to\_avg; i++){

status = Pozyx.doPositioning(&position, dimension, height, algorithm);

if(status == POZYX\_SUCCESS){

if(position.x>0 && position.y>0){

if(position.x<5001 && position.y<5001){

out\_x = out\_x + position.x;

out\_y = out\_y + position.y;

}else{

i--;

}

}else{

i--;

}

}else{

i--;

}

}

x\_loc = out\_x/num\_to\_avg;

y\_loc = out\_y/num\_to\_avg;

}

// get heading info updates global heading variable with what appears to be magnetic heading

void getHeading(){

euler\_angles\_t euler\_angles;

int status;

status = Pozyx.getEulerAngles\_deg(&euler\_angles, NULL);

if(status == POZYX\_SUCCESS){

//

heading = euler\_angles.heading;

}

}