

navigation.c

```
#include <stdio.h>
#include <stdlib.h>

float xGoal; // location of goal in meters from center of field
float yGoal; // location of goal in meters from center of field
float xRobot; // location of robot in meters from center of field
float yRobot; // location of robot in meters from center of field
float thetaRobot; // orientation of robot

// digital to analog conversions done to read voltage outputs from sensors
// dac values supplied by function that reads inputs in some other file

extern int dacOrientation; // orientation angle in tenths of a degree
// range from 0 to 360
// 0 to 180 is left of x-axis
// 180 to 360 is right of x-axis

extern int dacAccelForward; // acceleration reading in millivolts
extern int dacAccelLeft; // acceleration reading in millivolts

extern float sin3600(int); // defined in trig.c
extern float cos3600(int); // defined in trig.c

// accelerometer produces 290 millivolts per G
// gravitational constant G is 9.8 meters / second / second
#define MPSPS_PER_MILLIVOLT (9.8/290.0)
#define SEC_PER_UPDATE (1.0 / 30.0 )

void initNavigation() {
    xRobot = 0.0;
    yRobot = 0.0;
    xGoal = 1.0;
    yGoal = 2.0;
}

void updateNavigation() {
    // assume DAC produces a value with positive=forward, negative =backwards
    float aForward = dacAccelForward * MPSPS_PER_MILLIVOLT;
    float aLeft = dacAccelLeft * MPSPS_PER_MILLIVOLT;
    thetaRobot = dacOrientation * 0.1;

    // acceleration measured in relative to robot orientation
    // compute acceleration in coordinates of field
    float sinTheta = sin3600(dacOrientation);
    float cosTheta = cos3600(dacOrientation);
    float ax = aForward * cosTheta + aLeft * sinTheta;
    float ay = aLeft * cosTheta + aForward * sinTheta;

    static float vx = 0.0; // velocity in meters per second
    static float vy = 0.0; // velocity in meters per second

    vx = vx + ax * SEC_PER_UPDATE;
    vy = vy + ay * SEC_PER_UPDATE;

    xRobot = xRobot + vx * SEC_PER_UPDATE;
    yRobot = yRobot + vy * SEC_PER_UPDATE;

    printf("vx=%g, vy=%g\n", vx, vy);
    printf(" x=%g, y=%g, theta=%g\n\n", xRobot, yRobot, thetaRobot);
}
}
```