

## Programmers Notes & Homework #2

11/8/2006

Attached is a modified version of the code we worked up last night. The major change is that last night we ignored the fact that the robot might not be oriented in the same direction all the time. I also divided moved the navigation functions into a separate file. Navigation.c has the functions for dead reckoning and main.c has the code that calls the functions defined in navigation.c. In order for the whole thing to work we still need a third file I call trig.c that can compute sine and cosines.

Trig.c should contain the following code. Note where you need to complete the sin3600 and cos3600 functions. The argument to these functions is in tenths of a degree with legal values between 0 and 3599. The sin90 function takes two arguments one an integer number of degrees and another integer number of tenths of a degree. It only allows the number of degrees to be between 0 and 90 and the number of tenths to be between 0 and 9. Your job is to make use of the symmetry in the sine function and the fact that  $\cos(t) = \sin(t+90)$  to get the job done. So, the cos3600 function can use sin3600 to get its job done and the sin3600 can use the sin90 function to get its job done. Note that to convert from tenths to whole degrees you can just use  $\text{theta}/10$  since C will just discard the remainder. To get the tenths of a degree, you can do some fancy arithmetic or just use C's % operator to get the remainder  $\text{theta}\%10$  will be the remainder after division.

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

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

```
float sinTable[91] = {
```

```
    0.000000000000, 0.017452406437, 0.034899496703, 0.052335956243,  
    0.069756473744,
```

```
    0.087155742748, 0.104528463268, 0.121869343405, 0.139173100960,  
    0.156434465040,
```

```
    0.173648177667, 0.190808995377, 0.207911690818, 0.224951054344,  
    0.241921895600,
```

```
    0.258819045103, 0.275637355817, 0.292371704723, 0.309016994375,  
    0.325568154457,
```

```
    0.342020143326, 0.358367949545, 0.374606593416, 0.390731128489,  
    0.406736643076,
```

```
    0.422618261741, 0.438371146789, 0.453990499740, 0.469471562786,  
    0.484809620246,
```

```
    0.500000000000, 0.515038074910, 0.529919264233, 0.544639035015,  
    0.559192903471,
```

```
    0.573576436351, 0.587785252292, 0.601815023152, 0.615661475326,  
    0.629320391050,
```

```
    0.642787609687, 0.656059028991, 0.669130606359, 0.681998360062,  
    0.694658370459,
```

```

    0.707106781187, 0.719339800339, 0.731353701619, 0.743144825477,
    0.754709580223,

    0.766044443119, 0.777145961457, 0.788010753607, 0.798635510047,
    0.809016994375,

    0.819152044289, 0.829037572555, 0.838670567945, 0.848048096156,
    0.857167300702,

    0.866025403784, 0.874619707139, 0.882947592859, 0.891006524188,
    0.898794046299,

    0.906307787037, 0.913545457643, 0.920504853452, 0.927183854567,
    0.933580426497,

    0.939692620786, 0.945518575599, 0.951056516295, 0.956304755963,
    0.961261695938,

    0.965925826289, 0.970295726276, 0.974370064785, 0.978147600734,
    0.981627183448,

    0.984807753012, 0.987688340595, 0.990268068742, 0.992546151641,
    0.994521895368,

    0.996194698092, 0.997564050260, 0.998629534755, 0.999390827019,
    0.999847695156,

    1.000000000000

};

```

```

// weighted average of table lookups good to 4 decimal points

```

```

float sin90(int degrees, int tenths) {
    float s = sinTable[degrees];
    if (tenths == 0)
        return s;
    float s1 = sinTable[degrees + 1];

    return ((10-tenths)*s + tenths*s1) / 10;
}

```

```

float sin3600(int theta) {

```

```
    // insert code here
}

float cos3600(int theta) {
    // insert code here
}
```