PROJECT 2 REPORT

Michael Meli

ECE 561

# Optimizations – Inclination Measurement and Calculations

## Optimizations

The following table explains the optimizations performed and the corresponding execution time improvements:

|  |  |  |  |
| --- | --- | --- | --- |
| Optimization Performed | Profile Ticks (10,000 runs) | Improvement (in ticks) | Time Per Single Run (μs) |
| Initial (no optimization) | 10418 | ------ | 1041.8 |
| Force optimization for time and highest optimization level (O3) using compiler options | 10415 | 3 | 1041.5 |
| Set the --fpmode=fast (fast floating point mode) compiler option | 6674 | 3741 | 667.4 |
| Change the calculation of atan2() to atan2f() and sqrt() to sqrtf(), which operate on floating point data instead of converting to double | 3328 | 3346 | 332.8 |
| Changed the I2C clock speed to 1.2MHz using a SCL divider of 20 (0x00) | 2051 | 1277 | 205.1 |
| Square root approximation using online guide [4] given in the project spec | 1992 | 59 | 199.2 |
| Put parentheses around the expression “180/PI” to allow the compiler to precompute the value | 1955 | 37 | 195.5 |
| atan2 approximation using the polynomial approximations given by [1] and [2] in the project spec | 1538 | 417 | 153.8 |

As the table above shows, the final runtime of an inclination measurement and calculation was approximately 153.8 microseconds.

## Profiling Results at Each Step

The following tables list the top 5 functions in execution time profile for each of the above steps in the optimization process.

**Initial** (no optimization):

|  |  |
| --- | --- |
| Function Name | Ticks |
| i2c\_wait | 3143 |
| \_\_aeabi\_dmul | 2603 |
| \_double\_epilogue | 1191 |
| \_\_aeabi\_ddiv | 788 |
| \_\_aeabi\_llsl | 418 |

Forced optimization for time and highest optimization level (O3):

|  |  |
| --- | --- |
| Function Name | Ticks |
| i2c\_wait | 3192 |
| \_\_aeabi\_dmul | 2890 |
| \_double\_epilogue | 1070 |
| \_\_aeabi\_ddiv | 923 |
| \_\_aeabi\_llsl | 408 |

Setting the --fpmode=fast compiler option:

|  |  |
| --- | --- |
| Function Name | Ticks |
| i2c\_wait | 2499 |
| \_\_aeabi\_dmul | 1775 |
| \_double\_epilogue | 355 |
| \_dsqrt | 334 |
| \_\_aeabi\_dadd | 320 |

Changing the use of atan2() to atan2f() and sqrt() to sqrtf() to operate with floats:

|  |  |
| --- | --- |
| Function Name | Ticks |
| i2c\_wait | 2837 |
| \_fsqrt | 135 |
| \_\_aeabi\_fmul | 120 |
| \_\_aeabi\_fadd | 90 |
| \_\_aeabi\_fdiv | 65 |

Increasing the I2C clock speed to the fastest possible:

|  |  |
| --- | --- |
| Function Name | Ticks |
| i2c\_wait | 979 |
| \_\_aeabi\_fmul | 227 |
| \_\_aeabi\_fdiv | 124 |
| \_\_aeabi\_fadd | 123 |
| \_fsqrt | 102 |

Implementing a square root approximation:

|  |  |
| --- | --- |
| Function Name | Ticks |
| i2c\_wait | 909 |
| \_\_aeabi\_fmul | 376 |
| \_\_aeabi\_fadd | 204 |
| \_\_aeabi\_fdiv | 118 |
| i2c\_repeated\_read | 50 |

Put parentheses around (180/PI) to allow the compiler to precompute the value:

|  |  |
| --- | --- |
| Function Name | Ticks |
| i2c\_wait | 842 |
| \_\_aeabi\_fadd | 268 |
| \_\_aeabi\_fmul | 131 |
| i2c\_repeated\_read | 129 |
| \_\_aeabi\_fdiv | 124 |

Implementing an atan2 approximation (**final** results):

|  |  |
| --- | --- |
| Function Name | Ticks |
| i2c\_wait | 894 |
| \_\_aeabi\_fdiv | 401 |
| i2c\_read\_setup | 143 |
| i2c\_repeated\_read | 53 |
| read\_full\_xyz | 34 |

# Optimizations – Magnetometer Calculations to Determine Tilt-Compensated Heading

## Optimizations

The following table explains the optimizations performed and the corresponding execution time improvements:

|  |  |  |  |
| --- | --- | --- | --- |
| Optimization Performed | Profile Ticks (10,000 runs) | Improvement (in ticks) | Time Per Single Run (μs) |
| Initial (only optimizations were setting the –O3 highest optimization flag and enabling optimize for time) | 8750 | ------ | 875 |
| Set the --fpmode=fast (fast floating point mode) compiler option | 2098 | 6652 | 209.8 |
| Change the calculation of cos(), sin(), and atan2() to cosf(), sinf(), and atan2f(), which operate on floating point data instead of converting to double | 2056 | 42 | 205.6 |
| Reused values for repeated trigonometric functions to reduce recalculation of the same values | 1666 | 390 | 166.6 |
| atan2 approximation using the polynomial approximations given by [1] and [2] in the project spec | 1032 | 634 | 103.2 |
| Implemented a cos approximation using a second degree Taylor Series polynomial | 871 | 161 | 87.1 |
| Implemented a sin approximation using a third degree Taylor Series polynomial | 735 | 136 | 73.5 |

As the table above shows, the final runtime of a tilt-compensated heading calculation was approximately 73.5 microseconds.

## Profiling Results at Each Step

The following tables list the top 5 functions in execution time profile for each of the above steps in the optimization process.

**Initial** (only optimizations were setting the O3 highest optimization option and enabling optimize for time):

|  |  |
| --- | --- |
| Function Name | Ticks |
| \_\_aeabi\_dmul | 10203 |
| \_\_aeabi\_dadd | 2313 |
| \_double\_epilogue | 1469 |
| \_\_aeabi\_llsl | 1415 |
| \_\_aeabi\_ddiv | 1346 |

Setting the --fpmode=fast (fast floating point mode) compiler option:

|  |  |
| --- | --- |
| Function Name | Ticks |
| \_\_aeabi\_fmul | 652 |
| \_\_aeabi\_fadd | 320 |
| sinf | 204 |
| \_\_aeabi\_fdiv | 110 |
| cosf | 103 |

Changing the calculations of cos, sin, and atan2 to cosf, sinf, and atan2f to operate with floats:

|  |  |
| --- | --- |
| Function Name | Ticks |
| \_\_aeabi\_fmul | 954 |
| \_\_aeabi\_fadd | 501 |
| sinf | 144 |
| \_float\_round | 114 |
| cosf | 113 |

Reusing values of calculations that were performed multiple times:

|  |  |
| --- | --- |
| Function Name | Ticks |
| \_\_aeabi\_fmul | 424 |
| \_\_aeabi\_fadd | 241 |
| sinf | 119 |
| atan2f | 93 |
| \_\_aeabi\_fdiv | 59 |

Implementing an atan2 approximation:

|  |  |
| --- | --- |
| Function Name | Ticks |
| \_\_aeabi\_fmul | 421 |
| \_\_aeabi\_fadd | 278 |
| cosf | 84 |
| \_\_aeabi\_fdiv | 84 |
| sinf | 57 |

Implementing a cos approximation using a Taylor Series polynomial:

|  |  |
| --- | --- |
| Function Name | Ticks |
| \_\_aeabi\_fmul | 332 |
| \_\_aeabi\_fadd | 160 |
| sinf | 111 |
| calc\_tilt\_comp\_heading | 111 |
| \_\_aeabi\_fdiv | 80 |

Implementing a sin approximation using a Taylor Series polynomial (**final** result):

|  |  |
| --- | --- |
| Function Name | Ticks |
| \_\_aeabi\_fmul | 295 |
| \_\_aeabi\_fadd | 147 |
| \_\_aeabi\_fdiv | 101 |
| sin\_approx | 93 |
| calc\_tilt\_comp\_heading | 93 |

# Development Effort Tracking

**Estimated person-hours required:** 25 hours

**Actual person-hours spent:** 22 hours

A good deal of development time (around 10 hours) was spent attempting to fix the error where the I2C bus would lock up at high baud rates. Another approximately 5 hours was spent attempting to set up SPI communications with the V2Xe compass, which was unsuccessful. The rest of the time was spent optimizing code for speed.