
Review for Test #2

ECE 376 Embedded Systems

Please visit [Bison Academy](#) for corresponding
lecture notes, homework sets, and solutions

Test #2 Topics

C Programming

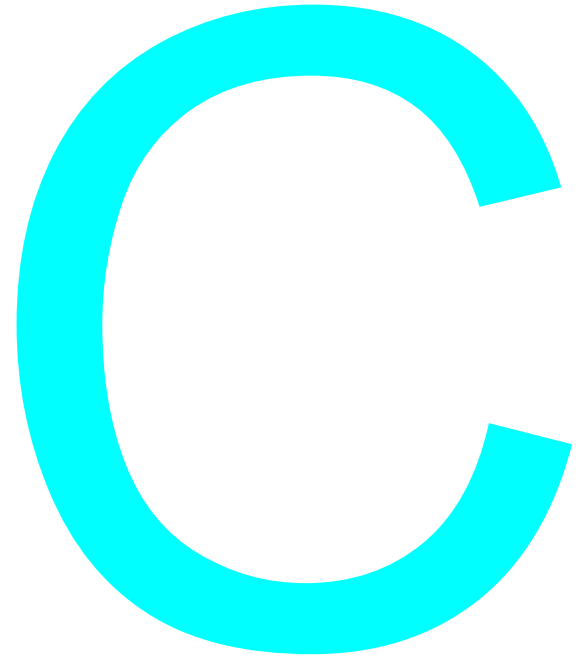
- C coding
- Binary I/O in C
- If-Statements
- For-Loops
- While-Loops
- Subroutines

Analog Inputs

- Reading the A/D
- Instrumentation Amplifiers

Statistics:

- Chi-Squared Test
- Student t-Test



C Programming

Rules in C

- Variables must be defined before being used
- Variable names must start with a character
- Single variable to the left of equals
- Terminate each line with a semi-colon

Optional

- Indentation is decorative
- Carriage returns are decorative

C Instructions

Valid:

```
unsigned int A1, A2, A3;
```

Invalid:

```
unsigned int 1A, 2A, 3A;
```

Valid

```
Y = 2*A + 3*B + 4*C;
```

Invalid

```
Y - 4C = 2*A + 3*B;
```

Variable Types in C

Name	bits	range
char	8	-128 to +127
unsigned char	8	0 to 255
int	16	-32,768 to +32,767
unsigned int	16	0 to 65,535
long	32	-2,147,583,648 to +2,147,483,647
unsigned long	32	0 to 4,294,967,295
float	32	3.4e-38 to 3.4e+38
double	64	1.7e-308 to 1.7e+308
long double	80	3.4e-4932 to +3.4e+4932

C Instructions

Name	Example	Operation
+	1 + 2 = 3	addition
-	3 - 2 = 1	subtraction
*	2 * 3 = 6	multiplication
/	6 / 3 = 2	division
%	5 % 2 = 1	modulus
++	A++ ++A	
--	A-- --A	
&	14 & 7 = 6	logical AND
	14 7 = 15	logical OR
^	14 ^ 7 = 9	logical XOR
>>	14 >> 2 = 3	shift right
<<	14 << 2 = 56	shift left

Binary I/O in C

Ports can be addressed 8-bits at a time

- Read: Read in all 8 bits
- Write: Write to all 8 bits

Bits can be addressed one bit at a time

Byte Commands

```
Data = PORTA;
```

```
PORTC = PORTB & 0x0F;
```

Bit Commands

```
Data = RB0;
```

```
RC0 = ! RC0;
```

If-Statements

Example C Code

Conditions

! not
= assign
== compare
> greater than
< less than
>= greater than or equal to
!= not equal

```
if (A == 3) {  
    PORTB = 2;  
    PORTC = 3;  
}  
else if (A == 4) {  
    PORTB = 5;  
    PORTC = 6;  
}  
else {  
    PORTB = 7;  
    PORTC = 8;  
}
```

Brackets indicate statements within the if-statement

- No brackets only execute to the semi-colon

For-Loops

Examples

For-Loops repeat a section of code a fixed number of times

- Starting point
- Do-while-true
- What to do after each loop

```
for(i=0; i<10; i++) {  
    PORTC = i;  
    for(j=0; j<1000; j++);  
}
```

Decrement: Count from 10 to 1

```
for(i=10; i>0; i--) {  
    PORTC = i;  
    for(j=0; j<1000; j++);  
}
```

While-Loops

Examples

White loops continue as long as the condition is true

Infinite Loop

```
while(1) {  
    :  
}
```

Wait until button RB0 is pressed

```
while(!RB0);
```

Wait until any button is pressed

```
while(PORTB == 0);
```

Subroutines in C

Example

Subroutines are sections of code which can be called from anywhere in a program

Subroutines must be defined before they are called

You can pass data to a subroutine

- These are received as local variables
- Changing them in the subroutine does not change the value in the main routine

You can return data from a subroutine

```
int Square(int X)
{
    int Y;
    Y = X * X;
    return(Y);
}
```

Flow Charts in C

Example

Like assembler, convert each block to a section of C code

- Makes grading easier
- Allows me to give partial credit

For-Loops

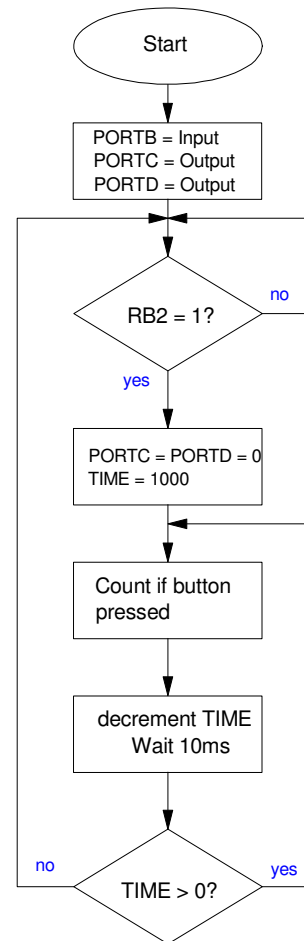
- Section is executing a fixed number of times

While-Loops

- Infinite loop
- Keep looping until a condition is met

Goto-Statements

- Avoid
- Implement using if/for/while loops



Timing in C

Example

Unlike assembler, executing time in C is not simply counting the number of lines of code

- Some instructions take one clock
- Some instructions take thousands of clocks

One way to measure time is to

- Set then clear an I/O pin, or
- Toggle an I/O pin

Using an oscilloscope, you can then measure the execution time

```
int Square(int X)
while(1) {
    RC0 = !RC0;
    A2D = A2D_Read(0);

    RC1 = 1;
    V = A2D * 0.000488;
    RC1 = 0;

    LCD_Move(0,0);
    LCD_Out(V*100, 4, 2);

    for(i=0; i<1500; i++);
}
```

Analog Inputs (Software)

Example

The PIC has 13 analog inputs

These can be read by

- Initializing the A/D port
- (include a section of C code)
- Call the routine A2D_Read(3);

The number passed is the A/D channel

- Returns 0 - 1023
- 0 = 0V
- 1023 = 5.00V

```
unsigned int A0, A1, A2;  
float V0, V1, V2;
```

```
// Read the raw A/D value  
A0 = A2D_Read(0);  
A1 = A2D_Read(1);  
A2 = A2D_Read(2);
```

```
// Convert to voltage  
V0 = A0 * 0.00488756;  
V1 = A1 * 0.00488756;  
V2 = A2 * 0.00488756;
```

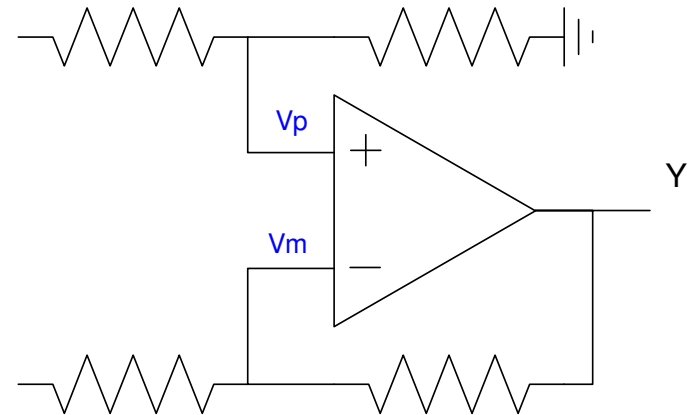
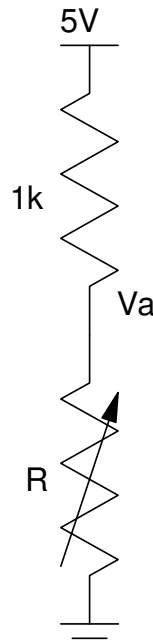
Analog Inputs (Hardware)

The A/D accepts 0V - 5V

Instrumentation Amplifiers allow you to convert an input voltage to 0V-5V range

Example: Output

- 0V @ $R = 800\ \Omega$
- 5V @ $R = 1000\ \Omega$



Statistics: Chi-Squared Test

Example

A chi-squared test is a test of a distribution

- Is a die fair?

Procedure:

- Collect data
- Separate the data into M bins
- Compare the expected frequency (np) to the observed frequency (N)
 - Compute the chi-squared score
- Convert to a probability using a chi-squared table
 - or StatTrek
 - Degrees of freedom = #bins - 1

Roll	p	np	N	chi-squ
1	1 / 6	10	8	
2	1 / 6	10	12	
3	1 / 6	10	7	
4	1 / 6	10	9	
5	1 / 6	10	13	
6	1 / 6	10	11	

Statistics: Student t-Test

A student t-test is a test of the mean

Example

Procedure:

- Collect data
- Compute the mean and standard deviation
 - Matlab can do this
- Find the t-score
 - Distance to the mean in terms of standard deviations
- Find the area of the tail
 - t-table
 - StatTrek also works
 - degrees of freedom = sample size minus one

Populations vs. Individuals

- Populations: Divide the standard deviation by the square root of the sample size

In the first six games of the year, the Vikings score {23, 24, 28, 28, 29, 24}

- *mean = 26.00*
- *standard deviation = 2.61*

What is the chance the Vikings will score more than 30 points in their next game?

- *Individual question*

What is the chance that the Vikings will average more than 30 points per game in an infinite game season?

- *Population question*
-

Test #2

4-5 questions

- Show your work (especially online)

In-Person

- During class time

Online

- BlackBoard
 - Available 6am - midnight
 - 75 minutes
 - Random order
 - No back tracking
-