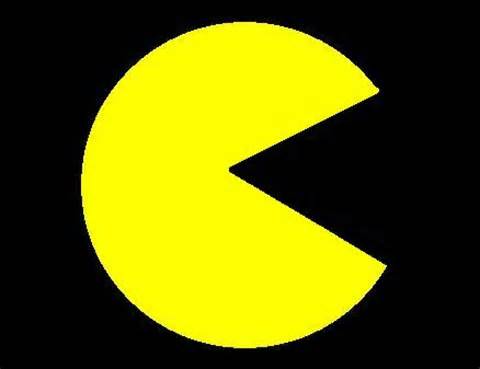
# EE/CPE 3280 Assignment 1

## Programming Exercise- Pacmania

In this exercise, you will become familiar with the basic C programming environment in PSoC Creator and develop an application that displays a Pacman image (as shown below) on the Graphical LCD (GLCD) display.



### PACMANia Functional Requirements

1. The Pacmania program shall display a “Pacman” image (as shown above) on the PSoC4 GLCD.
2. The wedge-shaped mouth of the Pacman image shall be made to appear to open and close repeatedly as long as the program is running.

### PACMANia IMPLEMENTation Requirements

1. The Pacmania program shall utilize the GLCD library provided by the instructor.
2. The implementation shall provide a C Function call to draw a line from a center point to a point defined using a polar coordinate system. The calling interface for this function shall be:  
    void GLCD\_Draw\_Line\_Polar(int centerX, int centerY, int angle, int length, int color);  
   An angle of 0 degrees is pointing straight to the right, and angles increase in a counterclockwise direction.  
   It is OK to put this in the main.c file, but make sure that the function definition is outside of main().
3. The implementation shall utilize the function GLCD\_Draw\_Line\_Polar to draw the required images.

### Implementation Notes

* Start by making a copy of the test program from assignment zero and just modify main.c
* In order to use trigonometric functions, you’ll need to use the C math library. There are two steps to include this:
  + Put the line #include <math.h> at the top of your main.c file with other includes.
  + Add the “m” library to the project:  
    In PsoC Creator Project|Build Settings|Linker|General Additional Libraries, add the library “m”
* You may find help on math functions by searching for “math.h c” on the Internet.
  + Trigonometric functions in math.h use radians. You’ll need to convert to/from degrees as needed.
  + The arguments and return values of trig functions are float or double.
* The GLCD library defines the X and Y axes as shown below (note that this is unconventional)  
  
* The GLCD library includes a function, GLCD\_Draw\_Line, that draws lines using Cartesian (x,y) coordinates. The center point for GLCD\_Draw\_Line\_Polar is already expressed in Cartesian coordinates. Your implementation of GLCD\_Draw\_Line\_Polar should convert the endpoint from polar to Cartesian coordinates and use GLCD\_Draw\_Line to do the drawing. Information on converting Cartesian to Polar coordinates can be found here: <http://en.wikipedia.org/wiki/Polar_coordinate_system>

## TO TURN IN Through Canvas

Turn in the following:

1. Submit a single MS-Word document containing the following through Canvas
   1. Top-level schematic (select all of your drawing, copy and paste it into the Word doc)
   2. main.c (Select all test with ctrl-a, copy and paste into the Word doc)
   3. Copy in any other files that your created or modified
2. Demonstrate your program to the instructor during class time on the due date. (Online students may send a video demo to the instructor)