EE 354   **Due: Monday, October 22, 2018**
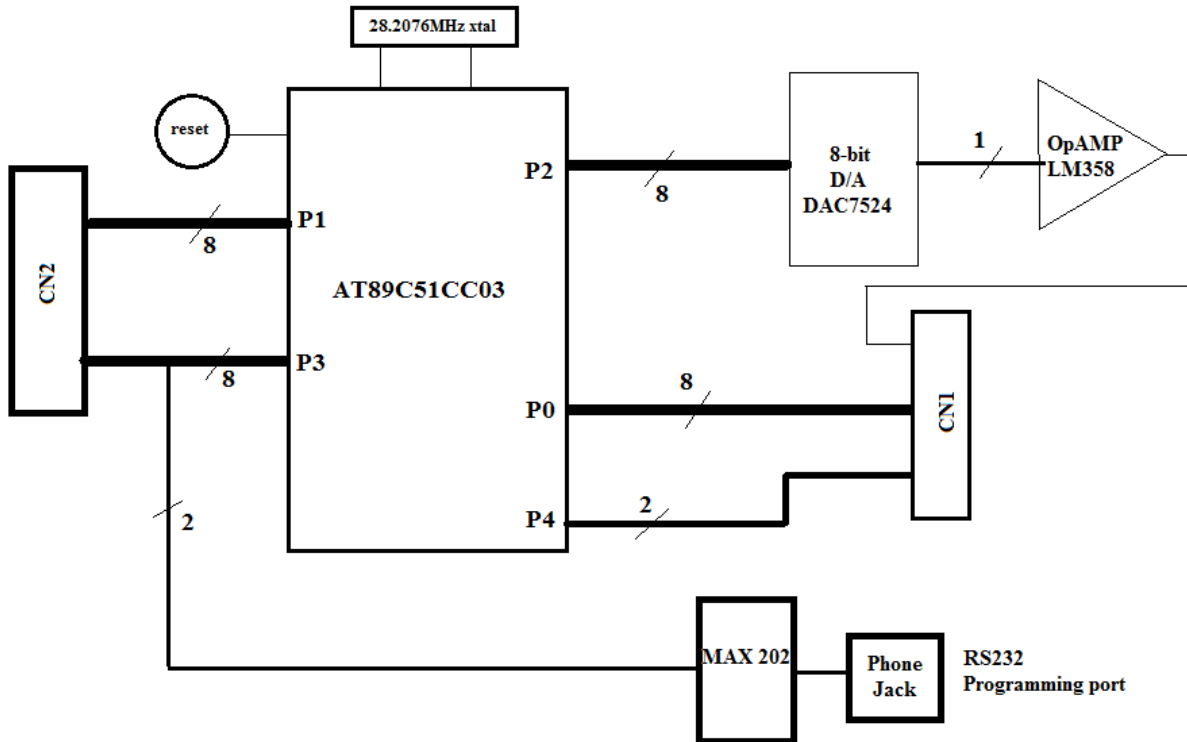**Project 1 Documentation Report**

The project report should consist of:
• A cover sheet with your name, the project number and title, and the date turned in.
• A list of novel features. Creativity may consist of novel hardware or software
   implemented features or a novel packaging technique.
• A list of those items you were able to demonstrate as working to the instructor.
• A discussion of how you considered safety, reliability, economic, manufacturability,
   and environmental factors. You need a few sentences describing what you did for
   each area.
• An estimate based on theoretical and empirical data as to the power requirements.
• Hardware documentation. At a minimum your hardware documentation must consist of
   a system diagram, a complete circuit diagram (with pin numbers), and a mechanical
   sketch or photo of your project done to a level of detail such that another person in
   the class could build your project from your diagram.

• Software documentation. At a minimum your software documentation should consist of
   fully commented source code for all of the modules in your program and a
   pseudocode design with enough detail that another person in the class could duplicate
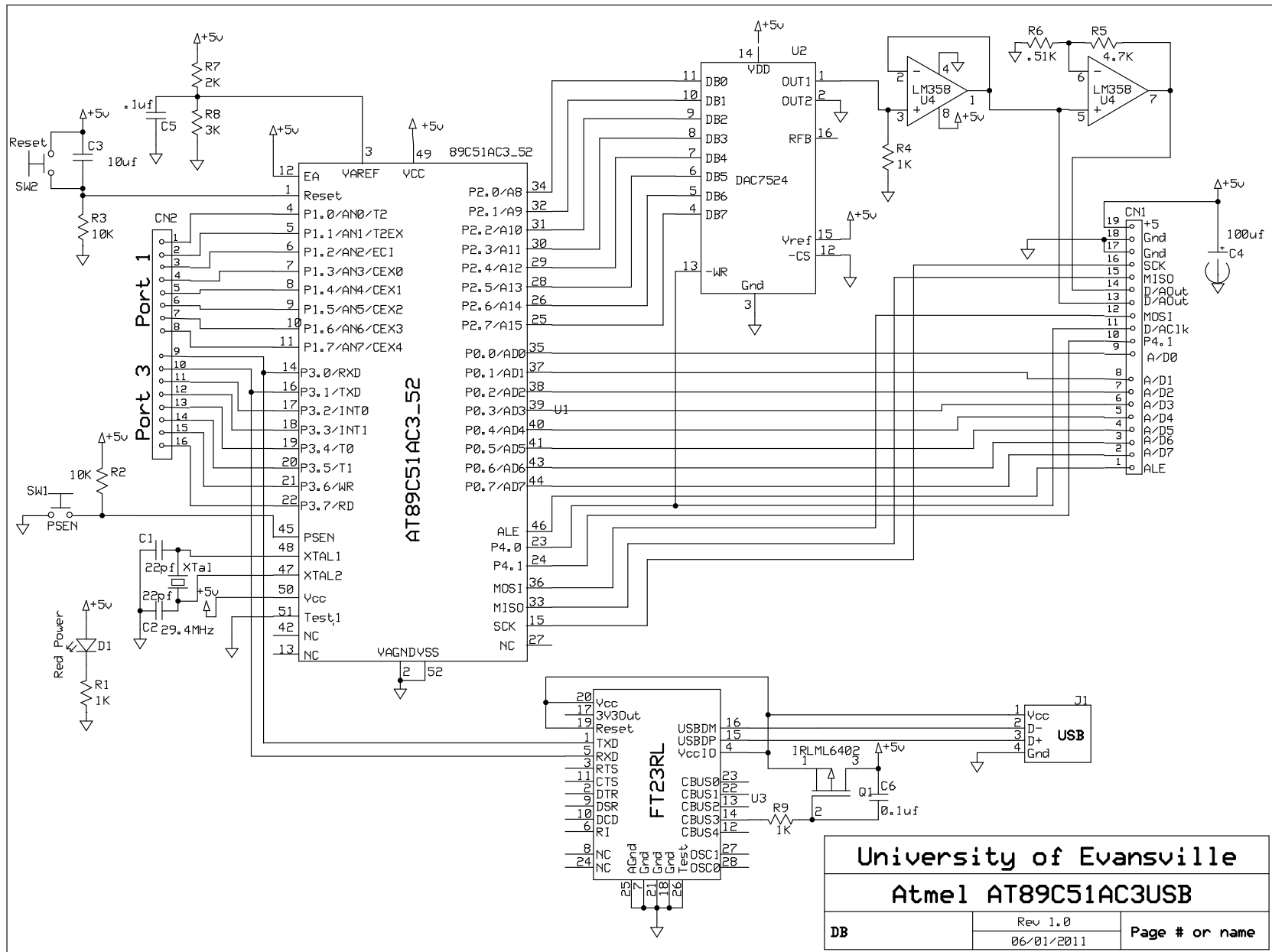   the function of your software. Your pseudocode must match your actual C-code.

EE 354
Hardware Diagram Samples

Sample 1: System Diagram for the AT89C51CC03 board

Sample 2: Circuit Diagram for the AT89C51CC03 Board.

University of Evansville

Atmel AT89C51AC3USB

| DB | Rev 1.0 | Page # or name |
| | 06/01/2011 | |

EE 354 Pseudo Code Sample

```
Initialize global interrupt variables for timer 0 multiplexing
Initialize global constants
   code unsigned char LEFT = xxx;
Initialize global variables
   unsigned char refreshRow = 0;  //row to be refreshed
   unsigned char LED[16];          //Memory map of LED matrix
Main code
  {Initialize timer 0 interrupt
   Set up timer 0 for multiplexing
   Set up timer 1 for ball speed
   Set up A/D converter for paddles
   Initialize variables
      Clear LED memory map
      Place paddles in memory map
      unsigned char ballDir     //0 = left 1 = right
      unsigned char ballRow, ballCol
   Start timer for multiplexing display
   while(game is not over)   //main program loop
     {Wait for game serve switch
      Start ball speed timer (timer 1)
      ballRow = 3;ballCol = 0;
      ballDir = 0   //left
      while(turn continues)
         {MoveBall(ballRow, ballCol, ballDir)
          if(ball is at paddle)
             {if ball misses paddle
                 {turn is over
                  update score
                  if(score at end)
                      announce winner
                      game over
                 }
               else
                   ballDir = ~ballDir
             }
          Check and set left paddle
          Check and set right paddle
          wait for ball speed timer to run out
         }
     }
  }
while(1);  Game is over wait for reset

Timer 0 multiplexor interrupt
  {reload the timer value
   Get the data d = LED[refreshRow]
   Convert refreshRow to port bits for decoders
   Make decoders active
   Send d to the 74LS244 port
   Update refreshRow
  }
```

Notes:
1. In this version the player always plays against the computer. The player always serves.
2. The ball only goes back and forth – not up or down. To get the ball to go up a row or down a row when it is hit you need to keep track of where on the paddle the ball was hit and move the ball up or down when it reaches the middle of the array.