# ENB244 Microcontroller Assignment

**1. Task:** Implement Pong and Breakout video games on the Teensy with the PCD8544 LCD screen. The game will be written in the C programming language. The human interface for the game will be the provided buttons, LEDs and LCD screen which will send and receive communication through the ATMEGA32's SPI communication port.

The assignment will be done individually or as a group of two (see section 7 for group specifications). You will demonstrate your game, explain the implementation and be asked a variety of questions about your game during a designated marking timeslot in week 13. The assignment code will be submitted online on Blackboard by **12:01pm Monday the 21st of October, week 13**.

You must submit your C programming code through Blackboard for plagiarism checking purposes. All submitted code will be compared through the use of plagiarism checker software. The penalty for plagiarism will be severe and handled according to the QUT plagiarism policy.

**3. Description:**

The game Pong is one of the earliest video games available to the general public. It was created in 1972 and spawned the initial success of Atari. The game simulates table tennis and consists of two paddles and a ball where both players attempt to get the ball past the opponent.

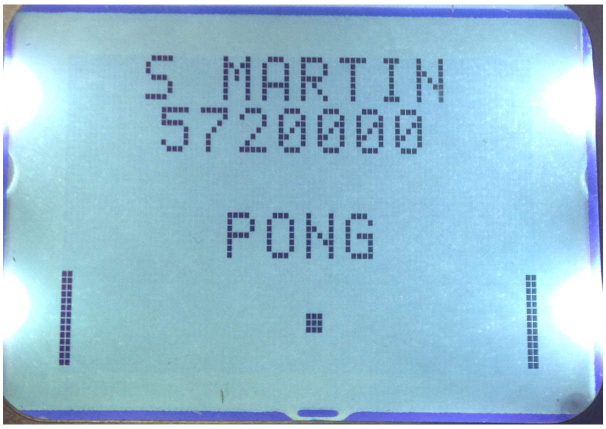
This game was extended in 1976 to become Breakout. This featured a similar style of gameplay except instead of playing an opponent; the ball was directed at bricks that were destroyed when the ball hit them.

The assignment consists of three parts: creating the game Pong, extending this to the game Breakout and implementing extended hardware functionality or game features.

**4. Implementation:**

## Pong – 50%

Below is a sample screen shots of a Pong game.



Your game should be a playable version of Pong that meets the minimum requirements below.

- Introduction screen
    - Must display Name, Student Number and Pong title. (1%)
    - Start the game by pressing a button. (1%)
    - Countdown to game start. (1%)
        - 3 – 2 – 1 @ 1Hz.
- Game screen should have
    - 2 paddles, 2x14 pixels at the left and right hand side of the screen. (2%)
    - A ball, 3x3 pixels. (2%)
    - The ball should start in the centre of the screen and move towards the Human player. (2%)
- Game physics
    - The ball should bounce off the top and bottom of the LCD screen (or "walls") of the game and should preserve the angle of travel. (5%)
    - The ball should change direction when it collides with a paddle. (5%)
    - The bounce angle should change proportionally to the distance from the centre of the paddle. (8%)
    - The speed of the ball should be controlled by the potentiometer coupled with the ADC. (5%)
- Control input
    - Buttons move one of the paddles up or down. (Human) (3%)
    - The second paddle should follow the ball when it moves up or down. (Computer) (3%)
- If the ball does not hit the paddle the opponent is then declared the winner
    - This message should be displayed on the screen and a LED should flash @ 2Hz. (1%)
    - The game should return to the intro screen when both buttons are depressed. (1%)
- The game must be playable (10%)
    - Design considerations
        - Speed range of the ball
        - Speed of the paddle
        - Maximum angle of bounce
        - The game must be winnable

These specifications are provided as a guideline and for marking. Your game should meet these requirements and be playable. Depending on your implementation, some aspects are expected to differ regarding the look and feel of your game.

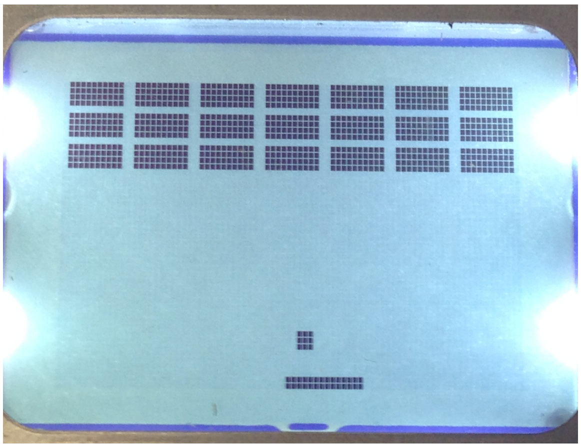
The game is to be able to be played by an unskilled user (your tutor). Based on this you should tailor the range of speeds as controlled by the ADC, the speed at which the buttons move the paddle, the rate at which the computer paddle moves and the maximum bounce angles. This is to ensure that the game is playable and can be won in a few minutes by a player.

## Breakout – 25%

You should extend your Pong game to incorporate the additional requirements.

- Start menu
    - Choose between Pong/Breakout (3%)
        - This could be done by making a select button and a start button.
- Game screen
    - One paddle.
    - One ball.
    - 7 columns of bricks, with 3 rows. (3%)
- The orientation of the game should be changed to horizontal instead of vertical as was implemented with Pong. (7%)
- Bricks should be destroyed when the ball bounces off them. (5%)
- The ball should bounce off the top and sides of bricks as they are destroyed. (4%)
- The game should end when you break all of the bricks. (3%)

Here is a screenshot of Breakout.



## Advanced Functionality – 25%

For the final 25% you may choose any following requirements that total to 25%. You may do more than 25% but no more than 25% will be given for this section. The difficulty of these tasks varies greatly - while you are encouraged to attempt the more challenging exercises you may also attempt several simpler tasks to receive the marks. **You must have completed all the basic functionality before attempting these tasks.**

- Additional Breakout levels (10%) *
    - You should add 5 additional levels to Breakout.
    - The levels should have varying patterns of blocks that increase in difficulty. At a minimum this should include at least one additional row per level.
    - You should advance levels when all the blocks are cleared.
    - You should add a cheat button code (press both buttons) to allow the levels to be won at any time for testing.

- Score (10%) *
    - You should add a scoring mode to both Pong and Breakout.
    - Pong should require the computer or Human to win 3 rounds before they are declared the winner.
    - For Breakout, you should receive a score based on the time taken to clear all the bricks.
        - You should add a cheat button code (press both buttons) to allow the levels to be won at any time for testing.

- Save high score in EEPROM (5%) *
    - You should add in functionality for the score from brick breaker to be saved in EEPROM.
    - This should be displayed on the start screen
    - This must be retained even when power is disconnected and reconnected.

- Additional lives (5%) *
    - You start Breakout with a total of 3 lives.
    - If you die the game restarts with one less life while preserving the remaining blocks.

- Set LCD backlight brightness (10%) *
    - Add additional options to the start menu to allow you to set the brightness.
    - This should give 10 levels of brightness form 0 to 100%.
    - This brightness should be saved in EEPROM and restored even after power loss.

- Sound (15%)
    - Obtain a speaker or Piezo buzzer from the Electronics Store on S block level 9.
    - Add interesting sounds to your game.

  - At a minimum you should have music played in the intro screen and sound effects when the ball collides with a paddle or brick.

- Add LED flicker when ball comes into proximity of bricks (10%)
  - Track the position of the ball relative to remaining bricks and each update cycle change the brightness of the LED based on the distance from the remaining bricks.
  - When the ball hits a brick the LED should be at maximum brightness.
  - After the ball hits a brick the LED should flicker (e.g. flash at 10Hz with decreasing brightness during on period).

- Power ups (15%)
  - When certain blocks in your game are broken these should produce a token that falls from the location of the block.
  - If your paddle collides with the token you receive an additional feature.
  - There should be at least 3 different power ups. For example:
    - Additional life
    - Wide Paddle
    - Fast mode
  - Where applicable these power ups should last for 15 seconds.

- Multiple balls for Breakout (10%)
  - Add an additional mode to Breakout where gameplay is for two balls instead of one.
  - The game should continue with only one ball.
  - You should restart or start the next level with two balls.

- Two-player Pong using potentiometer (5%)
  - The original Pong game used two potentiometers to control the position of the ball.
  - Add two player mode to your game using this pot to control the second humans position.
  - You should still be able to enter one player mode and the pot should still control the speed in this mode.

- Two player Pong using USB serial (25%)
  - Write a simple C program which communicates from your PC to the Teensy via USB serial (you cannot use Hyperterminal or any existing program).
  - This should not use any libraries that are not included with windows.
  - Define a simple protocol that allows you to control the second player paddle with from your PC.
  - The C code for your PC program must be submitted with your assignment.

- Secondary display on terminal (15%)
  - The terminal is a very powerful display environment.

  - The VT100 is an example of a terminal emulator which is supported by many programs.
  - This allows you to jump to positions on the terminal and write characters.
  - Using this, create a terminal 84 x 48 (Or the same size as your screen).
  - Echo the rendered screen to this terminal so it can be used as an additional display for gameplay.

- Falling blocks with Breakout (15%)
  - Add an additional challenge mode to Breakout.
  - In this mode the blocks fall at a constant rate.
  - If the blocks reach the bottom without all being broken the player loses.

**Notes**
* These tasks are expected to be the easiest to implement.

The additional functionality section is written to allow flexibility in your implementation and to make this assignment more interesting. If you fail to take this seriously or attempt to find loopholes that do not fulfil into the intended spec you will receive no marks. If you have questions about how some of the additional functionality should be implemented then please contact Steve or Liam.

## 5. Technical Details:

Your game must be written for the teensy and must compile using the AVR-LIBC environment. You may not use any additional libraries other than those which have been used in the prac classes.

## 6. Submission:

The assignment code will be submitted online on Blackboard by 12:01pm Monday on the 21st of October, week 13. This will then be downloaded, compiled and distributed to the prac tutors. In your marking slot you should bring along your hardware and your tutor will download the compiled hex file to your teensy and mark/play your game.

As this is an online submission the penalty for late submission is 10% per day of the total possible mark. Weekends count as two days.

Extensions will only be granted for medical, work or exceptional family circumstances. These require a medical certificate, letter from your employer, statutory declaration or other documentation. No other reasons will be considered. Any queries should be directed to Jinglan Zhang, jinglan.zhang@qut.edu.au .

**7. Group Variant:**

If you wish you may also complete this assignment as a group. In order to utilise all the hardware the following modifications to the above must be made.

- You should setup your screens so they are side by side on your breadboard.
- You should use UART or I2C communications between your two Teensys to synchronise the displays.
- These should update at the same time and your game should stretch across both screens.

  **This must work to receive any of the following marks.**

## Pong – 50%

- Introduction screen
  - Must display Name, Student Number and Pong title, Player number. (1%)
  - Start the game by both players pressing a button. (1%)
  - Countdown to game start. (1%)
    - 3 – 2 – 1 @ 1Hz. (Must be synchronised)
- Game screen should have
  - 2 paddles, 2x14 pixels at the left and right hand side of the respective LCD screens. (2%)
  - A ball, 3x3 pixels. (2%)
  - The ball should start in the centre of the two screens and move towards the right. (2%)
- Game physics
  - The ball should bounce off the top and bottom of the LCD screen (or "walls") of the game and should preserve the angle of travel. (5%)
  - The ball should change direction when it collides with a paddle. (5%)
  - The bounce angle should change proportionally to the distance from the centre of the paddle. (8%)
  - The speed of the ball should be controlled by the potentiometer coupled with the ADC. This should be on Player 1s Teensy. (5%)
  - The ball must preserve its direction and position when transferring between two screens.
- Control input
  - Buttons move each players paddle up and down. (Human) (3%)
    - There is no computer controlled paddle.
- If the ball does not hit the paddle the opponent is then declared the winner
  - This message should be displayed on the screen and a LED should flash @ 2Hz. (1%)
  - The game should return to the intro screen when both buttons are depressed by either player. (1%)
- The game must be playable (10%)
  - Design considerations
    - Speed range of the ball

        - Speed of the paddle
        - Maximum angle of bounce
        - The game must be winnable

## Breakout – 25%

You should extend your Pong game to incorporate the additional requirements.

- Start menu
    - Choose between Pong/Breakout (3%)
        - This could be done by making a select button and a start button.
        - This must be displayed on Player 1s teensy.
- Game screen
    - Two paddles. This should be constrained on each screen
    - One ball. This should move between screens
    - 7 columns of bricks, with 3 rows. (3%)
- The orientation of the game should be changed to horizontal instead of vertical as was implemented with Pong. (7%)
- Bricks should be destroyed when the ball bounces off them. (5%)
- The ball should bounce off the top and sides of bricks as they are destroyed. (4%)
- The game should end when you break all of the bricks. (3%)

## Advanced Functionality – 25%

For the advanced functionality you may not repeat the same advanced feature on more than one teensy. Each group member must implement a different set of features. You may also not implement the two player feature.

For example Teensy 1 may have scoring, EEPROM score storage and levels implemented in its code. And Teensy 2 may have sound and LCD Backlight setting.

**Submission**

Both group members should submit their versions of code to blackboard. This should be one set of code for each teensy. If you wish to attempt this as a group please email Liam O'Sullivan on lt.osullivan@qut.edu.au .

If you attempt this you attempt this you automatically receive a Gold Star.