Hung Tran

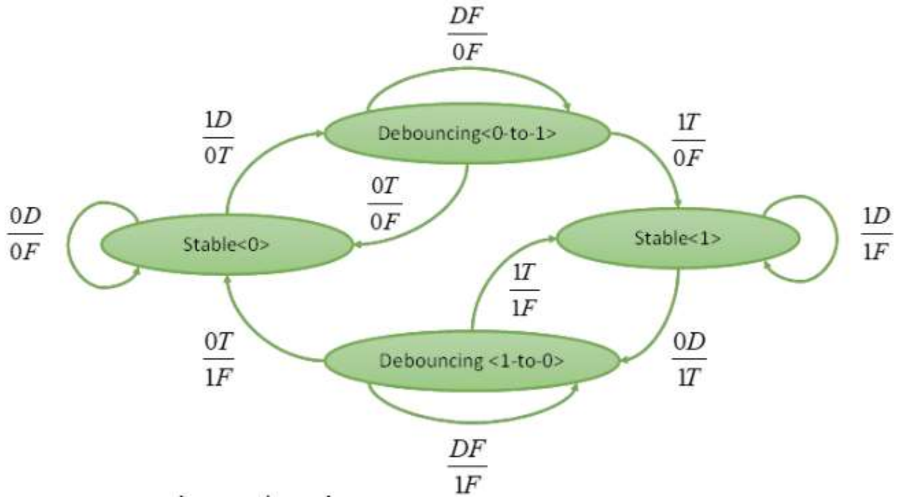
ECE 2534

Lab Report

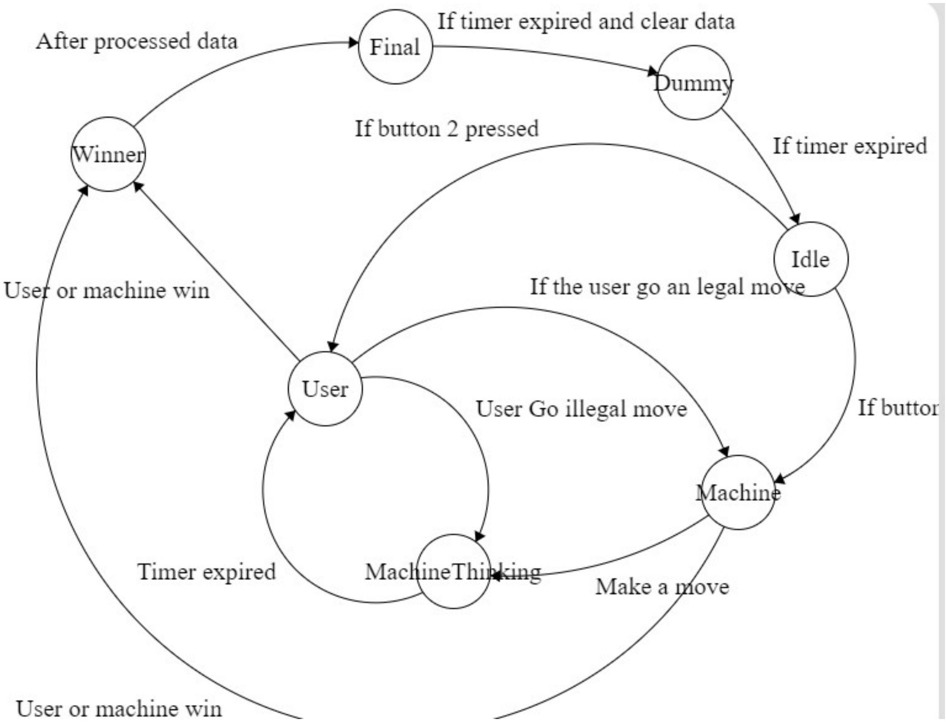
In this project, student create a game called “Tic Tac Tone”. This game is between user and computer. There are multiple state in the game. The “The mother” finite state machine will contain 4 states. The first state in idle more where the sample of the playing screen will appear. Every second, the cross and circle will move itself. On the top of the screen, there will be the game messages changing every 3 seconds. Those will be “TicTacTone”, “Score Computer - User”, “S1: I start”, “S2: You start”. The second state is game mode 1 which will make the computer goes first and game more 2 which let the user goes first. In game mode 1 and 2. There is another finite state machine which will be explained in the next paragraph. The last state will be give up state when the user press either button one or button two. Once in the give up state, the screen will be filled with cross and computer will win.

In the game mode finite state machine. There are 6 states which are idle, machine, user, winner, machine thinking, final, and dummy. In the idle state, it will wait for the button to press. If the button is one, idle state will go to machine state which make the machine goes first. If the button pressed is 2, idle state will switch to user case which allow the user to go first. In the machine state, it will go to the think state which let me computer think for a second then go back to the user state. There is a timer inside the thinking state. Once it goes to user state. It will start to listen to the tone. As soon as one of tone play, it will analyze it then see if it’s a illegal move or a good move. If it’s a illegal move, it will go to out the message illegal move then the state goes to machine thinking state then go back to user state. If it’s a legal move, it will go back to machine state. In every machine state and user state, the code will constantly check for winner. As long as there is a winner, either of state will go to the winner state. In the winner state, the screen will show up the message say either you win or you lose. In the winning state, it will move to the dummy state which there is a timer in there. It will wait for few second before going back to idle.

I also modify finite state machine of the button debounce. Once the button is press, it will only return a signal once.

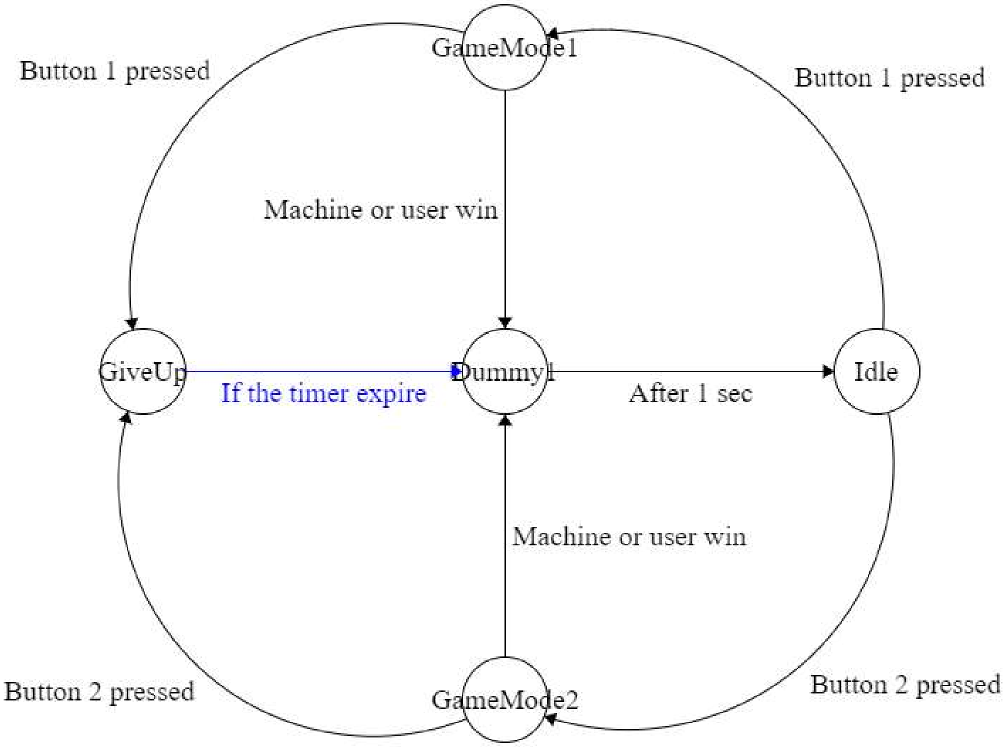


Finite State machine for button debouncing



Game Mode Finite State Machine

Mother Finite state machine



The code run successfully. All the function and feature works except for any extra credit.

Initialized the sound

```
void InitSound();
```

Play any given tone and time

```
void PlaySound(tnote n, unsigned ms);
```

play winning sound

```
void MachineSound(int i);
// This type presents the three possible states for a cell
typedef enum {empty, cross, circle} tcellstate;
// This function returns true when X wins
int      CrossWins(tcellstate map[9]);
// This function returns true when O wins
int      CircleWins(tcellstate map[9]);
// This function resets map sate to empty
void      ClearMap(tcellstate map[9]);
// This function fills all empty cells with 'X'
void      AbortMap(tcellstate map[9]);
// Adds a symbol v in a random empty location
void      RandomAdd(tcellstate map[9], tcellstate v);
// This function returns true of neither O nor X wins and no more moves are possible
int    Tie(tcellstate map[9]);
```

take in sound sameple and filter them

```
int  ScaleSample   (unsigned s);
```

convert the sample into power

```
void SampleGoertzel(Gtap *t, unsigned x);
```

reset the sound filter

```
void ResetGoertzel (Gtap *t);
```

get the power from the filter

```
int  PowerGoertzel (Gtap *t);
```

initial display

```
void InitDisplay();
```

draw the given message

```
void DrawMessage(char *s, uint32_t color);
```

draw the score

```
void DrawScore  (int computerscore, int humanscore, uint32_t color);
```

draw theplaying board

```
void DrawBoard  (tcellstate map[9]);
```

highlight the winner

```
void DrawWinner (tcellstate map[9],int winner, uint32_t color);
```

draw unsiged number in hex

```
void DrawUnsigned(unsigned line, char prefix[4], unsigned n);
```

draw a string

```
void PrintString(char *str, int row, int col);
```

set foreground color

```
void LCDSetFgColor(color_t c);
```

draw character on screen

```
void LCDDrawChar(unsigned row, unsigned col, int8_t c);
```

draw idle screen

```
void DrawIdle (int o3s,int Computer, int User);
```

initial top button

```
void InitButtonS1();
```

debound top button

```
int ButtonS1Pressed();
```

initial botton button

```
void InitButtonS2();
```

debounce second button

```
int ButtonS2Pressed();
```

check sample

```
int  ScaleSample   (unsigned s);
```