

ECE353 final project by Keren Zhu and Zeng Fan, Team No.07

ReadMe

1. Game introduction: Run Toad!
2. Comments and search key words
3. Real time communication configuration (Shake hands)

Game introduction: Run Toad!

Run Toad!

Game description:

Princess Peach is in danger again! Toad wants to inform Mario to rescue the Princess, but Bowser is on Toad's way. Toad needs to escape from Bowser, otherwise Mario will not come.



Control:

P1- Bowser: Press UP button to select Bowser (he is a black square in the game). Player needs to use joystick to move Bowser to catch Toad in a 3*3 matrix. Press RIGHT button to catch Toad when P1 is in the same square as Toad, but P1 only have 1 chance to catch Toad, otherwise Mario will come and beat P1.

P2- Toad: Press DOWN button to select Toad. Player needs to use joystick to move Toad to escape Bowser in a 3*3 matrix. Once Bowser fails to catch Toad, P2 wins. If Toad is caught, P2 loses.

Comments & search key words

ADC - PS2 Joystick

Configure ADC0 to use Sample Sequencer #2 to read the X and Y directions of the PS2 Joystick

Related files: ADC.c. Comments Name: CommentsADC.

ADC - Sample Sequencer #2

Sample Sequencer #2 should be configured to generate an interrupt after the second analog channel in the sequence has completed.

Related files: ADC.c. Comments Name: CommentsADC.

ADC - Timer0A Initiated

ADC conversions will be triggered using Timer0A interrupts

Related files: interrupts.c. Comments Name: CommentsADC.

UART0

Used for serial debug. 115200 baud with interrupts

Related files: boardUtil.c. Comments Name: CommentsUART

LCD - SPI Clk Speed

Communicate with the LCD at 25MHz

Related files: boardUtil.c. Comments Name: CommentsLCD

LCD - Text Strings

Display text strings

Related files: main.c. Comments Name: CommentsLCD

LCD - Graphics

Display non-trivial graphic images.

Related files: main.c. Comments Name: CommentsLCD

Wireless Communication - Tx and Rx

Transmit and receive on both devices

Related files: main.c. Comments Name: CommentsWireless

Wireless - Dropped/Sent Packets Total

Keep a running total of packets dropped/sent

Related files: main.c. Comments Name: CommentsWireless

Wireless - External Interrupts

Utilize external interrupt pin to detect received data

Related files: wireless.c. Comments Name: CommentsWireless

Timers - SysTick

Use SysTick timer with interrupts de-bounce push buttons

Related files: interrupts.c. Comments Name: CommentsSystick

Timers - Timer0A

Use Timer0A as a 32-bit count down timer with interrupts

Related files: timer0.c. Comments Name: CommentsTimer

Timers - Timer0A Interrupts

Timer0A initiates printing wireless packet statistics to the serial debug port every 5 Sec. Timer0A Interrupt is used to

Related files: main.c. Comments Name: CommentsTimer

Timers - Watchdog Timer

Have the watchdog timer halts the board if data has not been received from the remote board in the last 10 seconds

Related files: watchdog.c. Comments Name: CommentsWDT

EEPROM - Read Board Data

The following information should be read from the EEPROM and printed to UART0 when the MCU is reset –

Student 1: Your name - Student 2: Your partner's name - Team Number: xx If you do not know your team number, see the team assignments spreadsheet on the course website.

Related files: main.c. Comments Name: CommentsEEPROM

EEPROM - Write

When the PS2 pushbutton is pressed down, the user will be prompted to modify the contents of each of the three files. The user will be prompted to enter a new value. If the user presses enter, the current value is maintained. Pressing any other printable key or carriage return characters Each student name is limited to 80 ASCII characters. The team number should be stored

Related files: interrupts.c. Comments Name: CommentsEEPROM

EEPROM - Game Data

Read/Write stateful data such as a high score to the EEPROM

Related files: main.c. Comments Name: CommentsEEPROM

Push Buttons

Use directional push buttons connected to GPIOF

Related files: main.c. Comments Name: CommentsPB

Push Buttons - PS2

Pressing down (Z direction) on the PS2 joystick results in the user being prompted on UART0 to enter new information. Pressing the PS2 button down will stop the watchdog timer and place the board into a configuration mode that is used to modify the code. Pressing the PS2 button down will reset the board.

Related files: interrupts.c. Comments Name: CommentsPB

Creativity

Opening animation; real time multiplayer game; ending music

Completeness of Design

Yes

Establish connection: (Shake hands)

1. Player 1 (1P) sends 0x00 to 2P, 1P will retry until the 0x00 is sent successfully. Then

1P will wait for 0x01 or watchdog interrupts.

2. 2P will try to receive 0x00, when received, 2P sent 0x01 back to 1P, indicating the 2P is ready to start.

3. After receiving 0x01 from 2P, the 1P will send 0x02 to 2P to tell the 2P that the game should start.

4. After receiving 0x02, the 2P start the game. And when the game is running, he will continuously send 0xAX, which is the game information.

5. After receiving 0xAX from 2P, 1P knows that 2P has started the game, and 1P therefore also starts the game.

6. This mechanism ensure that the two player can start the game at the same time (within the interval of less than one frame).