QUEENSLAND UNIVERSITY OF TECHNOLOGY

CAB202 - MICROCONTROLLER ASSIGNMENT

SEMESTER 1, 2020

CHIRAN GAYANGA WALISUNDARA

N10454012

AIM

This assignment aimed to invest, design, implement, document and demonstrate the prototype of a microcontroller-based product concerning the learning outcomes of the unit CAB202. This product needs to perform a meaningful service, perform a specific useful function, or generally be a cohesive and sensible construct that delivers valuable functionality to the user.

INTRODUCTION

A prototype handheld game was virtually designed, developed and implemented using the TinkerCAD software named “GOBBLER RUSH”. This prototype product was targeted for the user to interact with the gaming environment and indulge in an enjoyable and interactive session similar to the ‘Pacman’ game but a smaller but faster scale. The game also has less decision making but requires quick reactions from the person playing it. If the player fails to do so, it counts as a loss of a life or the end of the game i.e. end of lives. The higher the score and survival time of the player, the better the player is at the game with higher reaction time.

WIRING INSTRUCTIONS

Wires – Each colour of wire represents a unique wire in terms of its functionality as given below,

1. Red wire – Connects with a component and supplies/caries 5V of voltage and current (power supply)
2. Black wire – Connects components with the Ground terminal ('GND' or '-' in Breadboard)
3. Purple wire – Connects the switch pins (PC1, PC2 & PC3) of the Arduino with terminal 2a (bottom rightmost terminal) of the switches on the Breadboard
4. Orange wire – Connects the Anode of the LEDs with the respective pins (PB2, PB3, PB4 & PB5) in the Arduino board
5. Turquoise wire – Connects the Data Input pins (DB4, DB5, DB6 & DB7) in the LCD screen with the respective pins in the Arduino board (PD4, PD5, PD6 & PD7).
6. Brown wire – Connects the Wiper (middle terminal) of the Potentiometer with pin PC0 of the Arduino board.
7. Yellow wire – Connects the 2nd leftmost 'LED' pin of the LCD screen with the 220 Ohm resistor.
8. Pink wire – Connects the 'RS' (Register Selector) pin of the LCD screen with pin PB1 of the Arduino board
9. Blue wire – Connect the 'E' (Chip Enabler Signal) pin of the LCD screen with pin PB0 of the Arduino board

Breadboard – A full-sized breadboard was used as the circuit connections require more space than the small breadboards. The '+' terminal marked on the top and bottom leftmost corners of the breadboard were connected to the pin on the left side of the Arduino Board. named '5V' where 5V power supply comes from the Arduino board using a wire (red) while the '-' terminal labelled o the breadboard was connected to the 'GND' terminal of the Arduino board (just below the 5V terminal) using a wire (black). The Breadboard used for the product has a vertical split in the middle which doesn't pass any Power or Ground supply which was connected to the top and bottom rightmost ends of the board. To overcome this barrier 2 wires were used to carry Power (red) and Ground (black) the left half of the Breadboard. All wires (red and black) from the LCD screen, switches, LEDs and Potentiometer are connected to the Breadboard to be supplied Power and Grounded.

Switches – 3 pushbutton switches were implemented into the game. Terminal 1a (bottom left terminal) of each switch was grounded while Terminal 2a (bottom right terminal) of each switch was wired (purple) on to the Arduino's respective pins allocated for switches (A0 to A5). In this scenario the switches from left to right of the breadboard ('Function button', 'Up button', 'Down button'), were connected to pins PC1(A1), PC2(A2) and PC3(A3) respectively on the Arduino board. A resistor of 1000 Ohms (1 kilo-Ohm) was also connected to Terminal 2a on one end and the other end of the resistor was connected to '+' terminal of the Breadboard (Power Supply) without the use of a wire.

LEDs – 4 LEDs were used for this application; From left to right in the breadboard, a Green LED, Red LED, Yellow LED and a Blue LED. The Anode (terminal to the right of the LED) of each LED was connected to pins in PORTB (D8 to D13) of the Arduino board using 4 separate wires (orange). Precisely pins PB2(D10), PB3(D11), PB4(D12) and PB5(D13) respectively. The Cathode (terminal to the left of the LED) of each LED was connected to the '-' terminal (Ground Terminal) of the Breadboard using a wire (black).

Potentiometer – Terminal 1 (leftmost terminal) of the Potentiometer was connected to the '+' terminal of the Breadboard (Power Supply) for it to have voltage and current flowing through and operate using a wire (red). The Wiper (middle terminal) of the Potentiometer was connected to pin PC0(A0) of the Arduino board using a wire (brown). Lastly, Terminal 2 (rightmost terminal) of the Potentiometer was connected to the '-' terminal of the Breadboard (Ground Terminal) of the Breadboard using a wire (black).

Resistors – Resistors of value 1000 Ohms (1 kilo-Ohm) were used for the switches and it connects terminal 2a (bottom right terminal) of the switch on one side with the '+' terminal (Power Supply) of the Breadboard on the other side. Another resistor was used in connecting the 2nd 'LED' pin of the LCD screen from the right on coming from the wire (yellow) one end with the '+' terminal (Power Supply) of the Breadboard on the other end. A resistance of 220 Ohms was used for this resistor.

LCD Screen – The LCD screen has 16 pins for wiring. 12 of the 16 were used to build the game. From the left, the 'GND' (Ground) pin of the LCD was connected to the '-' terminal (Ground Terminal) of the Breadboard using a wire (black). The 'V_{cc}' pin was connected to the '+' terminal (Power Terminal) of the Breadboard using a wire (red). The 'V_0' (Contrast Control) pin was also connected to the '-' terminal (Ground Terminal) of the Breadboard using a wire (black) as contrast control of any sort was not used in the game. The 'RS' (Register Selector) pin was connected to pin PB1(D9) of the Arduino using a wire (pink). The 'RW' (Read and Write mode) pin was connected the '-' terminal (Ground Terminal) of the Breadboard using a wire (black). The 'E' (Chip Enable Signal) pin was connected to pin PB0(D8) of the Arduino using a wire (blue). LCD pins 'DB0', 'DB1', 'DB2' and 'DB3' were unwired as the LCD screen was operating in 4 pin mode and not 8 pin mode. LCD pins 'DB4', 'DB5', 'DB6' and 'DB7' were connected to the Arduino board for pins PD4, PD5, PD6 and PD7 respectively using 4 separate wires (turquoise). The 1st 'LED' pin from the left of the LCD was connected to the 220 Ohm resistor using a wire (yellow) and the 2nd 'LED' pin from the left of the LCD was connected to the '-' terminal (Ground Terminal) of the Breadboard using a wire (black).

Arduino UNO Rev3 Board – The '5V' pin on the left of the Arduino board which supplies Power (Voltage and Current) and the 'GND' pin which is the Ground Terminal are connected to the Breadboard on the bottom leftmost slots. The '5V' pin was connected to the '+' Terminal of the Breadboard using a wire (red) while the 'GND' pin of the Arduino was connected to the '-' terminal of the Breadboard using a wire (black). This was Voltage and Current runs through the '+' terminal and Grounding runs through the '-' implementing it on any and every connection to the Breadboard. Given below is a summary of all the Arduino board connections and wires with their respective pins,

Arduino Pin	Wire	Connector
5V	red	Breadboard bottom left '+' slot
GND	black	Breadboard bottom left '-' slot
PC0(A0)	brown	Wiper (middle terminal) of Potentiometer
PC1(A1)	purple	'Function button' switch terminal 2a
PC2(A2)	purple	'Up button' switch terminal 2a
PC3(A3)	purple	'Down button' switch terminal 2a
PD4(D4)	turquoise	'DB4' pin of the LCD Screen
PD5(D5)	turquoise	'DB5' pin of the LCD Screen
PD6(D6)	turquoise	'DB6' pin of the LCD Screen
PD7(D7)	turquoise	'DB7' pin of the LCD Screen
PB0(D8)	blue	'E' pin of the LCD Screen

PB1(D9)	pink	'RS' pin of the LCD Screen
PB2(D10)	orange	Anode of the Green LED
PB3(D11)	orange	Anode of the Red LED
PB4(D12)	orange	Anode of the Yellow LED
PB5(D13)	orange	Anode of the Blue LED

LEARNING OUTCOMES

a. Digital I/O – switch

3 switches were used in the making of this product. Switches were used for the functioning of the game and for the movement of the gobbler. The switch to the leftmost side of the breadboard was named the 'Function button'. It was used to detect with a press if the player was done setting and selecting the number of lives the player wants to play the game with. The next two buttons to the right were named as 'Up button' and 'Down button' and as the name suggest it makes the gobbler move up and down while eating the berries on the screen.

b. Digital I/O – interrupt-based debouncing

Interrupt based switch debouncing has been incorporated on the switch named 'Function button' with the use of a timer (timer1) and successfully debounces it by detecting a single push of the button as clean and accurate input and outputs efficiently each time the game is played. The button takes into proper action when the player has to click it after adjusting the difficulty of the game the player wishes to play with.

c. Digital I/O – LED

4 LEDs (Green, Red, Yellow and Blue) have been used throughout the game. The LED from the leftmost side of the breadboard are named 'Green LED', 'Red LED', 'Yellow LED' and 'Blue LED' respectively. The Green LED lights up and lights down while the player plays the game and continues to light up till the player loses the game. The Red LED lights up at the start in the Welcome screen as well as when the player has to press the 'Function button' (which is below the Red LED). The Yellow LED and Blue LED light up in the welcome screen as well as when the controls to the game are displayed.

d. Analog Input

Analog input was implemented in a way that the player has to tune the Potentiometer (knob) to adjust the number of lives the player wishes to play the game with, options being (1 life, 3 lives or 5 lives). What actually happens here is the players input Analog value is taken in and an Analog to Digital Conversion (ADC) is done of the Potentiometers value with correspondence to the range of values in the Potentiometer to be tuned and the difficulty is thereby adjusted as the players preferences.

e. Analog Output (PWM) – hardware-based; OR software-based "bit-banging"

Software based PWM (Pulse-Width Modulation) was implemented with the use of a timer (timer0) to adjust the brightness of the Green LED which turns on when the gobbler starts eating the berries. Throughout the survival time of the player, the Green LED goes from no brightness to high brightness and back and forth gradually with the operation of 'bit-banging' (Data transmission process which uses software as a replacement for dedicated hardware for generating transmitted signals or processing received signals).

f. Serial I/O – hardware based; polling OR interrupt mediated

At the end of the game i.e. when the player has no lives to play the game with (lives = 0), the game ends and the time the player stayed alive in the game from the start till the end ('Survival Time') is printed on to the Serial Monitor in the form ('Survival Time: *number* seconds') after the score of the player is displayed. The time the player takes to play the game is calculated with the use of a timer (timer2) and interrupt.

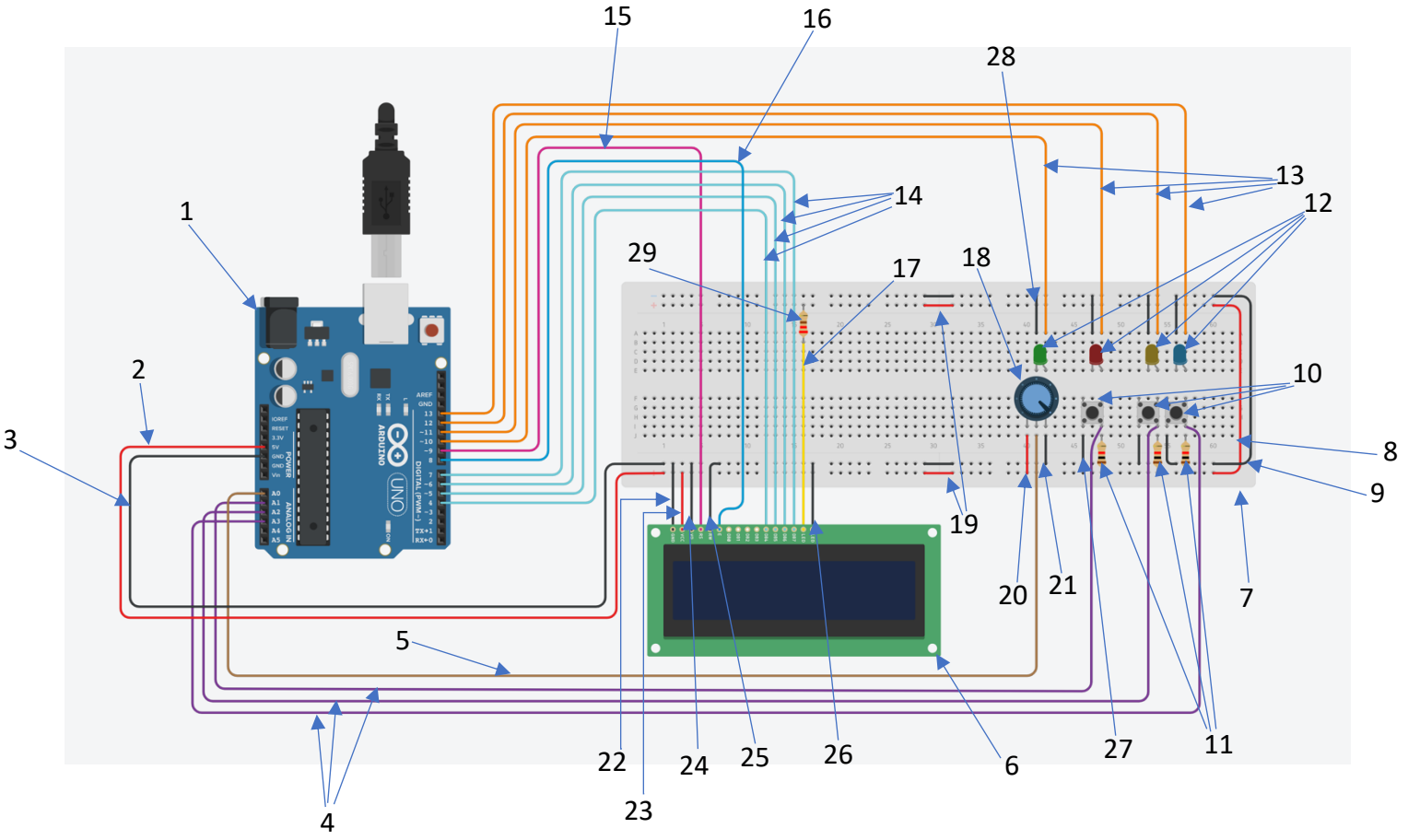
g. LCD – simple text display; OR: bit-mapped graphics, including sprites, line or pixel art

An LCD screen was used with 5 by 8 (5 columns & 8 rows) bit-mapped characters with pixel art and lines of text as well. At the start of the game, the LCD screen will display "Welcome to Gobblers Rush" and "Please Adjust Difficulty". Afterwards once he game actually starts, the Gobbler's open-mouth bit-mapped graphic will appear in close succession with the Gobbler's closed-mouth bit-mapped graphic to create an effect as if the gobbler is eating the berries. The berries were programmed default from the pixel art styles. The berries come at the gobbler one at a time where the LCD screen columns 0 to 12 scrolls with a small delay (very fast) to the left of the screen to make it seem as if the berries are moving towards the gobbler.

h. Timers – other than debouncing or PWM

The 3rd timer (timer2) was implemented to calculate the survival time of the player playing the actual game from start till the gobbler's lives end and can no longer continue to gobble the berries. The timer was implemented in a way such that every time the timer overflows, the overflown time was added on to a variable named 'Survival Time' so that it would actually build up real time (stopwatch implementation) and finally the time was recorded from the start of the eating to the end of the eating process and the players' 'Survival Time' is displayed on the Serial Monitor at the end of the game in seconds. The higher the time, the better the player is at the game with respect to the number of lives used.

FULLY LABELLED SCHEMATIC DIAGRAM



1. Arduino UNO Rev3 Board
2. Wire(red) connecting the '5V' pin of the Arduino with the '+' terminal of the breadboard
3. Wire(black) connecting the 'GND' of the Arduino with the '-' terminal of the breadboard
4. Wires(purple) connecting the pins PC1(A1), PC2(A2) and PC3(A3) with terminal 2a of switches form left to right ('Function', 'Up' and 'Down') respectively
5. Wire(brown) connecting pin PC0(A0) with the Wiper (terminal 2) of the Potentiometer
6. Liquid Crystal Display (LCD) → 16×2
7. Breadboard
8. Wire(red) connecting the bottom '+' terminal of the Breadboard with the top '+' terminal
9. Wire(black) connecting the bottom '-' terminal of the Breadboard with the top '-' terminal
10. 3 Switches: - 'Function button', 'Up button' and 'Down button' from left to right
11. 3 of $1k\Omega$ resistors
12. 4 LED's: - 'Green LED', 'Red LED', 'Yellow LED' and 'Blue LED' from left to right
13. Wires(orange) connecting pins PB2(D10), PB3(D11), PB4(D12) and PB5(D13) to the Anode (right terminal) of the LED's
14. Wires(turquoise) connecting pins PD4(D4), PD5(D5), PD6(D6) and PD7(D7) to the pins DB4, DB5, DB6 and DB7 in the LCD screen respectively
15. Wire(pink) connecting the pin PB1(D9) of the Arduino with the RS (Register Select) pin of the LCD screen
16. Wire(blue) connecting the pin PB0(D8) of the Arduino with the E (Enabler) pin of the LCD screen
17. Wire(yellow) connecting the 1st 'LED' pin of the LCD screen from the left with the 220Ω resistor
18. Potentiometer

19. Wires (red and black) connecting the two halves of the breadboard, left and right to overcome the barrier
20. Wire(red) connecting terminal 1(leftmost terminal) of the Potentiometer with the '+' terminal of the Breadboard
21. Wire(black) connecting terminal 2(rightmost terminal) of the Potentiometer with the '-' terminal of the Breadboard
22. Wire(black) connecting the 'GND' pin of the LCD screen with the '-' terminal of the Breadboard
23. Wire(red) connecting the 'V_{cc}' pin of the LCD screen with the '+' terminal of the Breadboard
24. Wire(black) connecting the 'V_0' pin of the LCD screen with the '-' terminal of the Breadboard
25. Wire(black) connecting the 'RW' (Read / Write) pin of the LCD screen with the '-' terminal of the Breadboard
26. Wire(black) connecting the 2nd 'LED' pin of the LCD screen from the left with the '-' terminal of the Breadboard
27. Wire(black) connecting terminal 1a (bottom left terminal) of the switch with the '-' terminal of the Breadboard
28. Wire(black) connecting the Cathode (left terminal) of the LED with the '-' terminal of the Breadboard
29. 220Ω resistor

LINK TO TINKERCAD SIMULATION

https://www.tinkercad.com/things/9L99ybvXnVi-cab202-assignment-final/editel?sharecode=MLtHn__CxYJ2NYW991xYSE3HuBZSnNDRZHkH0wXu4EE

LINK TO RECORDED AUDIO-VISUAL SCREEN-CAST

https://youtu.be/0In1Ejwq1aI

***by the time this video was uploaded it was on standard definition quality only, while high definition was still processing as the video was 168 megabytes.**