

COMP2121 Project

Description:

In this project you will develop a safe-cracking game to be played on the AVR development board. The system will need to maintain game state, take input from the player, and provide audio-visual feedback.

Your system should satisfy the following requirements. Marks will be allocated amongst the points listed in the following section. More marks will be given to more difficult features. Some bonus marks will be awarded for extensions beyond the spec.

Core requirements: (80%)

Start Screen

On device startup, the LCD screen should display the following text:

2121 16s1
Safe Cracker

The device should wait for the user to start the game using the left push-button (PB1).

Start Countdown Screen

After the push-button is pressed, the game should count down for three seconds.

The following message should be displayed:

2121 16s1
Starting in ?...

Where '?' should be replaced with the number of seconds remaining.

When the timer expires the game should proceed to the 'Reset POT' screen.

Reset POT Screen

On this screen, the player is prompted to return the potentiometer to zero in order to begin the round. This is done by turning the potentiometer dial (located between the push buttons) as far as it will go anti-clockwise.

The following message should be displayed on the screen:

Reset POT to 0
Remaining: ?

Where the number of seconds left is displayed in place of the '?'. The initial number of seconds is determined by the difficulty settings.

When the player has returned the POT to zero, and it has stayed there for at least 500ms, the game proceeds to the 'Find POT Pos' screen.

If the player doesn't return the POT in time, the game will instead go to the 'Timeout' screen.

Find POT Pos screen

On this screen the user is instructed to find the correct potentiometer position. This position must be randomly chosen in order to be secret each time the game is played.

Find POT Pos
Remaining: ?

The timer will continue counting down from the previous screen, without resetting. If it expires the game will proceed to the 'Timeout' screen.

The measured potentiometer voltage will be in the range from 0x000 to 0x3FF and will increase as the dial is turned clockwise.

The player can complete this screen by turning the potentiometer to the correct position without going past it. Going too far will make the game jump back to the 'Reset POT' screen without resetting the timer.

If the player is within 16 raw ADC counts under the target position this is considered correct. When the dial is held in this position for 1 second the game will proceed to the 'Find Code' screen and the countdown will stop.

Indicator LEDs

While on the 'Find POT' screen, the proximity to the correct position is indicated by the LED strip.

When within 16 raw ADC counts, all LEDs should be lit.

Otherwise when within 32 raw ADC counts, all except the top LED should be lit.

Otherwise when within 48 raw ADC counts, only the bottom 8 LEDs should be lit.

Otherwise all LEDs should be off.

Find Code Screen

On this screen, the player must identify the secret key on the keypad. The key must be chosen randomly so that it is secret each time the game is played.

The player searches by pressing each key on the keypad. While the correct key is pressed the motor should turn on and spin at full speed. The motor should stop if the key is released. To complete the screen, the correct key should be held down for 1 second.

The display should show the following:

Position found!
Scan for number

The countdown does not apply to this screen.

Round Completion

After the 'Find Code' screen is complete, the game will either enter a new round or enter a final screen.

If less than three rounds have been completed, a new round should start at the 'Reset POT' screen with a different set of secret positions and numbers. The timer is reset when going back to this screen.

If the third round has just been completed, the game will continue to the 'Enter Code' screen.

Enter Code Screen

This screen prompts the user to enter the 3-digit secret code found by completing the 'Find Code' part of the previous rounds.

The following text should be displayed on the LCD screen.

Enter Code

The countdown timer should not be active on this screen.

As each key is pressed, an '*' should appear on the bottom row of the LCD, starting from the left side.

If an incorrect digit is entered, the bottom line of the display must be cleared and the user must start entering the code again from the beginning. Holding down a keypad button must result in only a single digit being registered.

If the user successfully enters all three digits in order, the game should continue to the 'Game Complete' screen.

Game Complete Screen

The final screen should display the following message:

Game complete
You Win!

The game should stay on this screen until any keypad or push button is pressed, then it should return to the 'Start' screen.

The strobe LED should flash at a rate of 2Hz while on this screen.

Timeout Screen

This screen should display the following message:

Game over
You Lose!

The game should stay on this screen until any keypad or push button is pressed, then it should return to the 'Start' screen.

Advanced Features

Sounds

The system should use the provided mini speaker to add sounds to the game.

The speaker should be connected between the pin labelled PB3 and the pin labelled GND, in the top-right corner of the board. Despite the labelling, this pin is actually controlled by changing the value

of PORTB pin 0, or PB0. The sound may be generated by toggling the pin to produce a square wave. The frequency of the square wave should be as soothing as possible.

The speaker should be used to beep for 250ms every time the countdown timer decrements. It should beep for 500ms when a new round begins. It should beep for 1s when the player reaches the 'Timeout' or 'Game Complete' screens.

LCD Backlight

When on the 'Start', 'Timeout' or 'Game Complete' screens, the LCD backlight should turn off if no keys have been pressed for 5 seconds. It should turn on if any keys are pressed.

When turning on or off, the LCD backlight should fade smoothly over 500ms.

When on any other screen the LCD backlight should stay on.

Quit Button

If, during gameplay, the PB0 push button is pressed, the game should immediately stop and return to the 'Start' screen.

Difficulty Settings

While on the 'Start' screen, pressing the keys 'A' through 'D' should select one of the four difficulty options. The options are:

Key	Countdown
'A'	20
'B'	15
'C'	10
'D'	6

The difficulty may be displayed while on the start screen however you like.

Submission Information

You will need to submit the following items:

1. A soft copy of your complete source (all .asm files). Your program should be well commented.
2. A hard copy of your user manual. The user manual describes how a user plays your game, including how to wire up the AVR lab board. Make sure you indicate which buttons perform each action and how the LED and LCD displays should be interpreted.
3. A hard copy of the design manual. The design manual describes how you designed the game system. It must contain the following components:
 - a. System flow control. This component describes the flow control of the system at the module level using a diagram.
 - b. Data structures. This component describes the main data structures used in the system.
 - c. Algorithm descriptions. This component describes the main algorithms used in the system.

- d. Module specification. This component describes the functions, the inputs and the outputs of each module.

Overall, a person with knowledge about the subject and board should understand how your system is designed after reading this design manual.

Be sure to clearly specify your name, student ID, and lab group on all submitted documentation.

Demonstration

You will need to demonstrate your working project to an assessor on Thursday or Friday of week 13. Demonstration time slots will be determined closer to the due date.

You will need to submit the hard copies of the above documents during the demonstration. You will also need to bring a copy of your source code on a flash drive or similar.

Marking Scheme

This project is worth 100 marks. The marking scheme will be as follows:

- System Implementation (80 marks)
- Design Manual (10 marks)
- User Manual (5 marks)
- Coding style and commenting (5 marks)

Hardware

For reference, the following C code will trigger an ADC read from the potentiometer connected to PK8. This can trivially be translated into AVR assembly.

```
ADMUX = (3 << REFS0) | (0 << ADLAR) | (0 << MUX0);
ADCSRB = (1 << MUX5);
ADCSRA = (1 << ADEN) | (1 << ADSC) | (1 << ADIE) | (5 << ADPS0);
```

When the ADC read is complete the ADC interrupt will be triggered, and the measured value can be read from the ADCH:ADCL I/O registers.

Random Numbers

The easiest way to generate random numbers on the AVR chips is to utilize the random timing of a user pressing a button. If timer3 is running, then reading from TCNT3H:TCNT3L will give the current counter value. Since this counter wraps around fairly quickly (depending on the prescaler setting) reading from it as soon as the user presses a button will effectively give a random 16-bit value.