COVID-19 SOCIAL DISTANCE CHECKER

Microprocessors & Digital Systems

CAB202 Assignment:

Student Name: Rona Mae Rosal

Student Number: N10360387

Introduction

As the number of cases of CoVid-19 around the world continuously increase, the World Health Organization (WHO) announced a set of rules that needs to be implemented to prevent further big outbreaks.

According to the 7News Australia (2020), Australia was quick to act on the early stages of the pandemic and decided to close its borders last March 20, 2020. However, cases within Australia continue to fluctuate. While the implementation of safety protocols such as wearing of masks, social distancing, and isolating are in place there are still cases or events where people fail to follow these safety protocols, one example of this is being in crowded places such as rallies, pubs, and public transportation.

Due to this, I decided to implement a device that may help people go on with their daily lives with a more accurate measurement of social distancing. The Covid-19 social distance checker is an automated device worn or carried by the user and makes an alert whenever another person stands less than from the carrier. This will increase the level of safety measure of the user by creating less contact with people that may unknowingly have the virus.

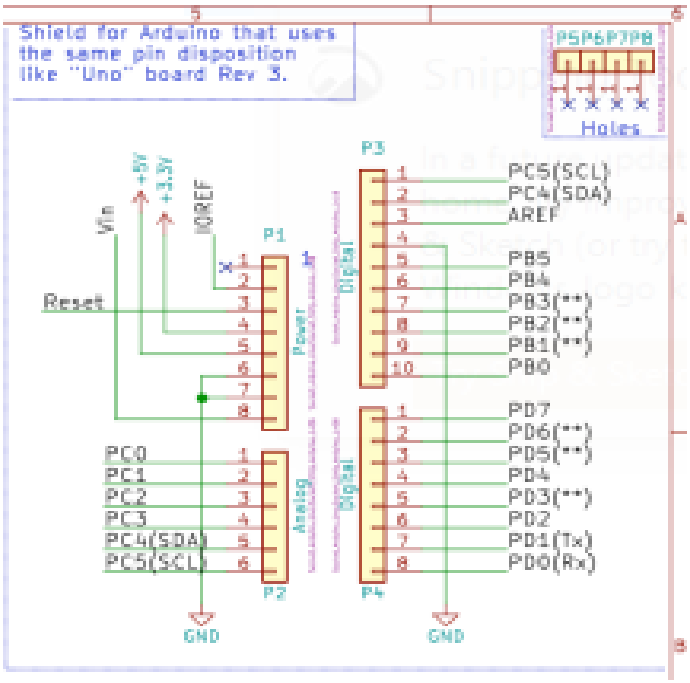
Upon the start of the application, it will trigger an overflow timer-based interrupt and display time count on the LCD screen as well as prompt a blinking green LED indicating that the routine is being used, within the interrupt service routine holds a condition indicating whether a button is pressed, a delay-based debouncing is introduced inside the ISR with a 50 milli-second delay. The program will properly commence upon pressing the left-most button and it will display a welcome message and instruction for the user to click the middle button to start detecting within a 12-inch radius. The sensor will then commence in detecting if a person is within the 12-inch margin using UART and will display a warning message along with a red blinking LED and trigger a warning beeping sound using the speaker to remind the user that social distancing must be practiced. Otherwise, a yellow blinking LED will signal the user along with a safe message indicating that social distancing is properly being practiced.

Learning Outcomes:

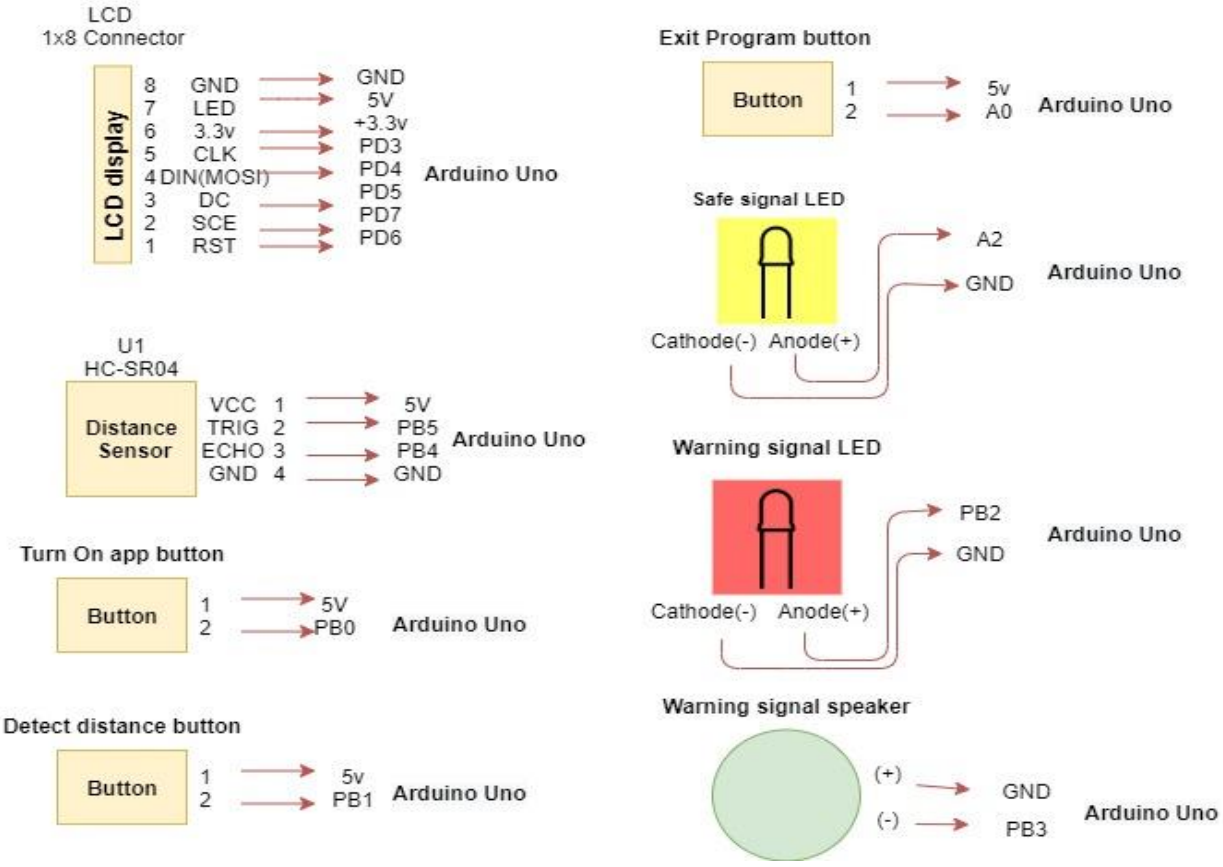
Digital I/O – Button	• **Turn on app button** - Button that calls turn_on_app function and starts the program. • **Detect distance button** - Button that activates the SC-HR04 distance sensor. • **Exit program button** – terminates the program.
Digital I/O – LED	• ***Safe signal LED (Yellow)*** - LED which glows when the sensor detects that a person/object's distance is more than 244 from the user, it prompts a safe signal that shows a person is following social distancing rules. • **Warning signal LED** - A red coloured LED which glows when the sensors detects that a person/object's distance is less than 244 from the user, it prompts a warning signal that shows a person is not following social distancing rules.
LCD	• **Message Display** – *Prints all kinds of messages and instructions for the users. The LCD initiates a welcome message upon turning on the application and signals instruction on how to detect a person or an object.*
Serial I/O – Polling	• *A for loop method constantly checks the function turn_on_app and checks if any changes on a button has been made before performing another desired function, such as detecting the distance.*
Digital I/O – Interrupt-based Debouncing (UART)	• *Upon the compilation of the program, it starts a timer overflow which triggers an interrupt and a delay debouncing is implemented within the Interrupt service routine when a button is pressed.*
PWM output	• *When the sensors detects that a person is within the 12-inch margin a warning sound using the speaker will be prompted to make a beeping sound along with a warning message on the LCD screen and a blinking LED light.*

Schematic

Arduino Uno



Hardware



Wiring Instructions & Components

- **Arduino Uno**
- **Breadboard**

  Used to build test circuit designs. Moreover, many holes into which circuit components like LED's and resistors can be inserted.

  Wiring Instructions

  1) Using a wire, connect the ground pin (GND) of the Arduino to the negative (-) power rail of the breadboard.
  2) Using a wire, connect the power supply (5V) of the Arduino to the positive (+) power rail of the breadboard.

- **LED**

  Wiring Instructions

  - **Safe LED**
    1) Using a resistor, connect the cathode (-) to the negative power rail of the breadboard.
    2) Using a wire, connect the anode (+) to the A2 of the Arduino Uno.
  - **Warning LED**
    1) Using a resistor, connect the cathode (-) to the negative power rail of the breadboard.
    2) Using a wire, connect the anode (+) to the PB2 of the Arduino uno.

- **Button**
  - **Turn on app button.**

  Wiring Instructions

  1) Using a wire, connect the first leg of the push button to the power rail (+) of the breadboard which is directly connected to the power supply (5v) of the Arduino uno.
  2) Using a resistor, connect the opposite leg of the push button to the negative rail (-) of the breadboard which is directly connect to the ground (GND) of the Arduino uno.
  3) Connect a wire under the same terminal strip as the resistor and connect it to the PB0 of the Arduino uno.

  - **Detect distance button.**

    Wiring Instructions

1) Using a wire, connect the first leg of the push button to the power rail (+) of the breadboard which is directly connected to the power supply (5v) of the Arduino uno.
2) Using a resistor, connect the opposite leg of the push button to the negative rail (-) of the breadboard which is directly connect to the ground (GND) of the Arduino uno.
3) Connect a wire under the same terminal strip as the resistor and connect it to the PB1 of the Arduino uno.

- **Exit program button**.

  Wiring Instructions

1) Using a wire, connect the first leg of the push button to the power rail (+) of the breadboard which is directly connected to the power supply (5v) of the Arduino uno.
2) Using a resistor, connect the opposite leg of the push button to the negative rail (-) of the breadboard which is directly connect to the ground (GND) of the Arduino uno.
3) Connect a wire under the same terminal strip as the resistor and connect it to the A0 of the Arduino uno.

- **Message Display (LCD)**

  Wiring Instructions

1) connect the ground pin (GND) of the LCD, to the GND pin of the Arduino Uno, this has a voltage of 0 and serves as a reference point to other electronic parts of the Arduino Uno.
2) connect the LED of the LCD to the positive (+) of the breadboard, which is directly connected to the power supply (5v) pin of the Arduino Uno, this serves as a back light to the LED when turned 1.
3) connect the 3.3v of the LCD to the +3.3v of the Arduino Uno. Voltages applied here is regulated at 3.3v.
4) Connect the serial clock pin (CLK) of the LCD to PD3 of the Arduino Uno, this signals that a bit is available on DIN.
5) Connect data serial input pin (DIN(MOSI)) of the LCD to the PD4 of the Arduino Uno, this transfers data to the LCD one bit at a time and interpret it as pixels,if D/C == 1 and interprets it as a command if D/C == 0.
6) Connect the data/command pin (DC) of the LCD to PD5 of the Arduino Uno, this pin interprets 0 as a command and interprets 1 as incoming data.

7) Connect the chip select pin (SCE) of the LCD to PD7 of the Arduino Uno, this tells the pin that there is incoming data.
8) Connect the reset pin (RST) of the LCD to PD6 of Arduino Uno, this pin resets the LCD to its default configuration when switched to 0.

- **Distance Sensor checker**

  Wiring Instructions

  1) Using a wire, connect the ground pin (GND) of the distance sensor to the negative (-) rail of the breadboard, which is directly connected to the GND of the Arduino uno.
  2) Using a wire, connect the 'Echo' pin of the distance sensor to the PB4 of the Arduino Uno
  3) Using a wire, connect the 'Trig' pin of the distance sensor to the PB5 of the Arduino Uno.
  4) Using a wire, connect the 'Vcc' pin of the distance sensor to the positive (+) rail of the breadboard, which is directly connected to the power supply (5V) of the Arduino Uno.

- **Warning signal Speaker**

  1) Connect the positive wire to the ground (GND) of the Arduino uno.
  2) Connect the negative wire to PB3 of the Arduino uno.

References:

Burke, K. (2020). Australia closes borders to stop corona virus. Retrieve from https://7news.com.au/lifestyle/health-wellbeing/australia-closes-borders-to-stop-coronavirus-c-752927