ENG1013 Milestone 1 Submission

Team A21

Authors: Connor Palios (34977716)
Ilija Trajkovski (34977015)
Huy Le (34990712)
Varon Rasiah (34550283)
Tevin Mudalige (34977635)

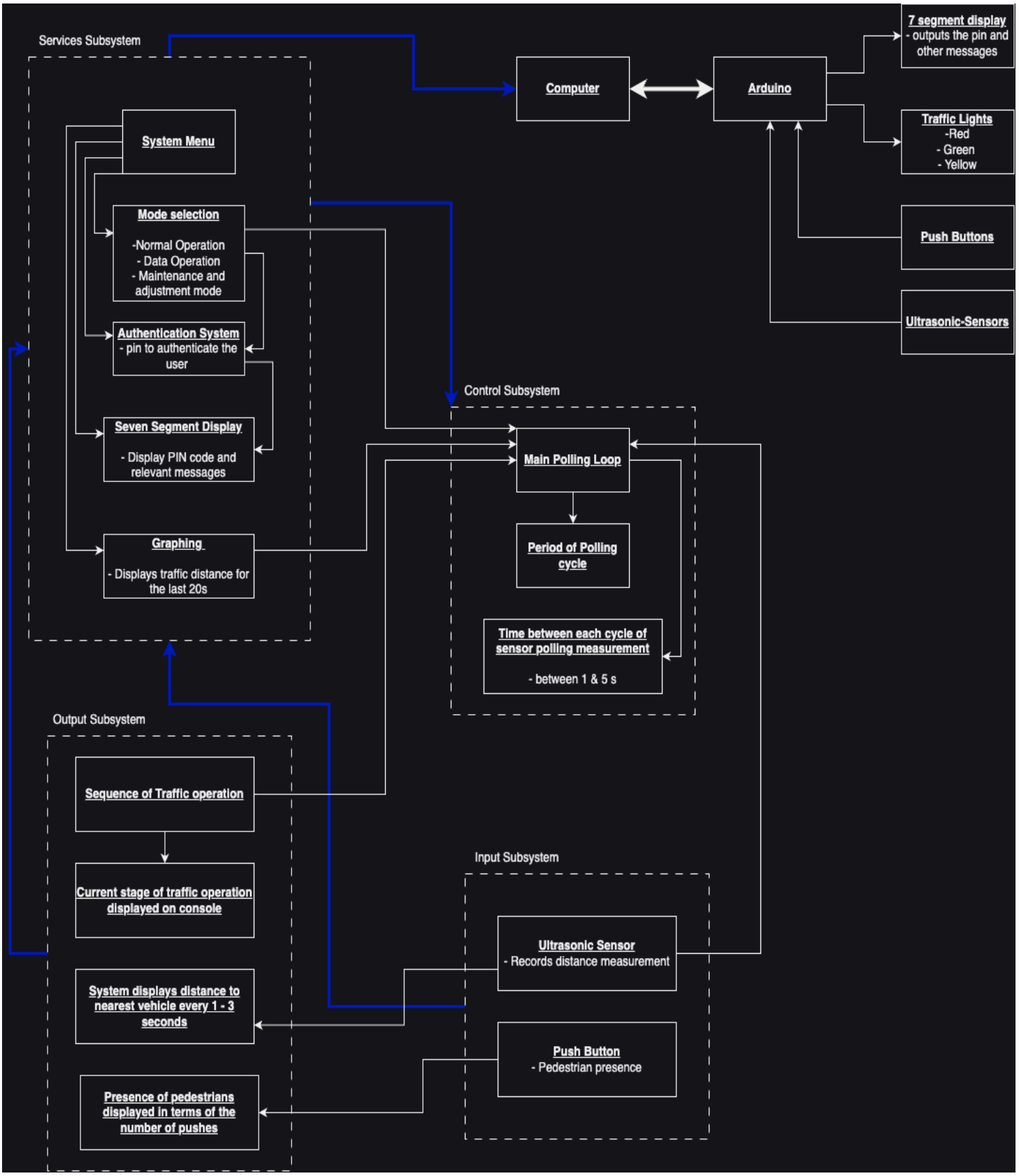
Manager: Bill Le

Table of contents

System Interaction Block Diagram

Index

- Blue arrows represent subsystem level connections
- White arrows represent regular connections between components



Subsystem One

Main/Script <Poll>:

1. **Call the services subsystem when the code is run.**

Subroutine/Function <Service>

Description: This is the function that contains the services sub-system that links all the subsystems together and provides an interface to the user for mode selection and also permits users to edit system variables and settings if required.
Inputs: <user-pin>
Outputs: <graphing, displays pin in the 7 segment display, displays relevant messages>
Function:

1. Initialise a subroutine called ‘Function Services ()’
2. Initialise a loop that ends with a keyboard interrupt
3. Check for whether there are any exceptions in the code, such as invalid formats or data types
4. Ask the user for an input, the computer should display “Please enter what mode you would like to select. 1 for Normal mode, 2 for Data observation mode, 3 for Maintenance and adjustment mode, and 4 to Discontinue the loop: “
5. Check whether the value entered by the user was between the values 1 and 4
6. If the value is not between 1 and 4, the computer should display “Please enter a value between 1 and 4 to choose from the selected modes”, then repeat from step 4
7. If value is within the range exit validation stage
8. The input that is put in by the user is defined to a variable “modeSelection”
9. Check if the variable “modeSelection” is equal to 1
    a. If it is equal to 1, the computer should directly call the control subroutine and send in “normal” as a arguments
10. The computer will check if the variable “modeSelection” is equal to 2
    a. If it is equal to 2, the computer should directly call the control subroutine and send in “data” as a arguments
11. The computer will check if the variable “modeSelection” is equal to 3
12. If so, will be the user will immediately be prompted to enter a 4 digit pin to be able to access the maintenance and adjustment mode
13. Create a variable to count the amount of times the user enters the pin, called ‘pinEntries’ and initialise this variable to 0
14. Initialise a loop that keeps looping until the pinEntries variable has a value of more than 3
15. The computer should display a message asking the user to enter the 4 digit pin to access the maintenance and adjustments mode
16. Display the 4 digit pin entered on the 7 segment display once it is entered by calling subroutine pin_entry and sending in the pin as an argument.
17. Increase the value of the pinEntries by 1
18. Check if authenticatePin equals pin which is defined as a system variable
19. In the case in which ‘authenticatePin’ equals true, grant access to the maintenance and adjustment system (Grants access to system variables and system variable editing)
20. In the case in which ‘authenticatePin’ equals false, do not grant access and ask the user to enter pin again
21. Once value of pin entries exceeds 3, lock the user out of being able to access the system
22. The computer will check if the variable “modeSelection” is equal to 4
    a. If it is equal to 4, the sequence should exit the loop, terminating the program

Subroutine/Function <Authenticate>

Description: authenticates the the pin entered by the user based on the pin stored in the

Inputs: user entered pin
Outputs: boolean of either true or false
Function:

1. Initialise a function named Authenticate that takes in a pin as an argument
2. Retrieve the password from the
3. If the pin is equal to the ‘system pin’ return the boolean as True
4. If not, a boolean value of False is returned.

Subroutine/Function <pin_entry>

Description: this function takes in the pin as an argument, reads every single element of the string based pin, converts and displays all the values of the pin to the 7 segment display.

Inputs: user entered pin
Outputs: data to the seven segment display
Function:

1. Initialise a dictionary named AlphaNumChars which contain the binary values needed for displaying characters and numbers on the 7 segment display.
2. Create a loop that loops on to every single element contained within the argument named pin
3. Use the value of the singular element obtained as the key in the dictionary and send in the corresponding value to the 7 segment display to be displayed.
4. Do this for all elements and pin and exit the loop.

Subsystem Two

Subroutine/Function <Control>

Description: This is the function that contains the main polling loop of the system, and saves the values that are taken by the ultrasonic sensor to dedicated variables. The values of the variables will decide the stage in which the traffic lights are set to. The control subsystem will also record the starting time and current time for which the system was running from. The control subsystem will also record the amount of times the pedestrian button has been pressed and will use this when taking into account the rotation and order of the traffic light rotation.

Inputs: Mode-selection choice from the user derived from the services subsystem.
Outputs: No outputs
Function:

1) Initialise the subroutine named control which takes in an input named mode.
2) Check which mode is specified

3) If the mode selected is named “normal” then
4) Call the Output Subroutine to initialise the traffic sequence
5) Initialise a block that looks for exceptions after this point in the code
6) Initialise the polling loop such that it terminates under a keyboard interrupt and runs based on the polling frequency.
7) Save time that the loop started in to startTime
8) Call subroutine Input to get current values for the number of button presses and save it in variable named buttonPresses
9) Call subroutine Input to get current values for the ultrasonic sensor 1 and save it in variable named sensor 1
10) Call subroutine Input to get current values for the ultrasonic sensor 2 and save it in variable named sensor 2
11) Run conditional statements based off of the above retrieved inputs and send changes to the output subroutine. This could either be a reduction in the time between stages or the skipping of stages based on the situation.
12) Save the current time in currentTime
13) Display the difference between currentTime and startTime as polling time to the user.

14) If the mode selected is named “data” then
15) Call subroutine Input to get current values for the number of button presses and save it in variable named buttonPresses
16) Call subroutine Input to get current values for the ultrasonic sensor 1 and save it in variable named sensor 1
17) Call subroutine Input to get current values for the ultrasonic sensor 2 and save it in variable named sensor 2
18) Display all the values of the variables obtained above periodically to the user in the form of a graph with appropriate axes and legends.

Subsystem Three

Subroutine/Function <Inputs>

Description: The input system is responsible for how data is recorded on the traffic light via ultrasonic sensors that are used to measure the distance of the coming cars as well as implementing push buttons to indicate the presence of pedestrians.

Inputs: data from ultrasonic sensors 1, ultrasonic sensors 2 and the pushButton
Outputs: list of ultrasonic sensors 1 data ,list of ultrasonic sensors 2 data, list of button pushes
Function:

1. Initialise a subroutine named “inputs” that returns a list of data and takes in an argument of “input type”.
2. Initialise empty lists named pushButton, sensor1 and sensor2
3. If the input type is equal to the push button (pedestrian button)
4. Then retrieve the recent data from the pushButton
5. Add this data to a list named pushButton.
6. Return the list
7. Else if the input type is equal to the sensor 1
8. Then retrieve the recent data from the sensor 1
9. Add this data to a list named sensor1.
10. Return the list
11. Else If the input type is equal to the sensor 2
12. Then retrieve the recent data from the sensor2
13. Add this data to a list named sensor2.
14. Return the list

Subsystem Four

Subroutine/Function <Output>

Description: This is the function responsible for the sequential operation of the traffic lights through all its stages. In addition to this the function also displays the number of push button presses periodically and also data from the ultrasonic sensor that holds distance based data to the user.

Inputs: list of Button presses, list of ultrasonic sensor 1 interactions, list of ultrasonic sensor 2 interactions
Outputs: Traffic operation stages, Graphing data, Distances to cars, number of button presses (pedestrian)
Function

1) Initialise subroutine named Output
2) Repeat until an interrupt or condition is required
3) Implement Stage 1 for 30 second and display Stage 1 on to the screen
4) Call subroutine input every 3 second and get the distance reading and store it in a variable named distance
5) Round this value to 2 decimal places
6) Display distance to screen
7) Implement Stage 2 for 3 second and display Stage 2 on to the screen
8) Call subroutine input and get the distance reading and store it in a variable named distance
9) Round this value to 2 decimal places
10) Display distance to screen
11) Implement Stage 3 for 3 second and display Stage 3 on to the screen
12) Call subroutine input and get values of button presses and display.
13) Implement Stage 4 for 30 second and display Stage 4 on to the screen
14) Call subroutine input every 3 second and get the distance reading and store it in a variable named distance
15) Round this value to 2 decimal places
16) Display distance to screen
17) Implement Stage 5 for 3 second and display Stage 5 on to the screen
18) Implement Stage 6 for 3 second and display Stage 6 on to the screen
19) Call subroutine input and get the distance reading and store it in a variable named distance
20) Round this value to 2 decimal places
21) Display distance to screen
22) Repeat above steps from step 2 until condition is met.