MIDDLE EAST TECHNICAL UNIVERSITY

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

EE447 LABORATORY PROJECT

EGE EKİN CANSIZ 2231561

Contents

1 Overall Project

1.1 Project Definition

In this project i build a car park sensor system with GUI using different components and Tiva C Launchpad Board. These components are:

```
Nokia 5110 LCD
Tiva C Launchpad
Nokia 5110 LCD
HC-SR04 Ultrasonic Sensor
Stepper Motor
5KOhm Potentiometer
```

In the next part i will explain each module's firmware.

1.2 Explanatory Video and Photos

https://www.youtube.com/watch?v=SCHFnKnwRsc

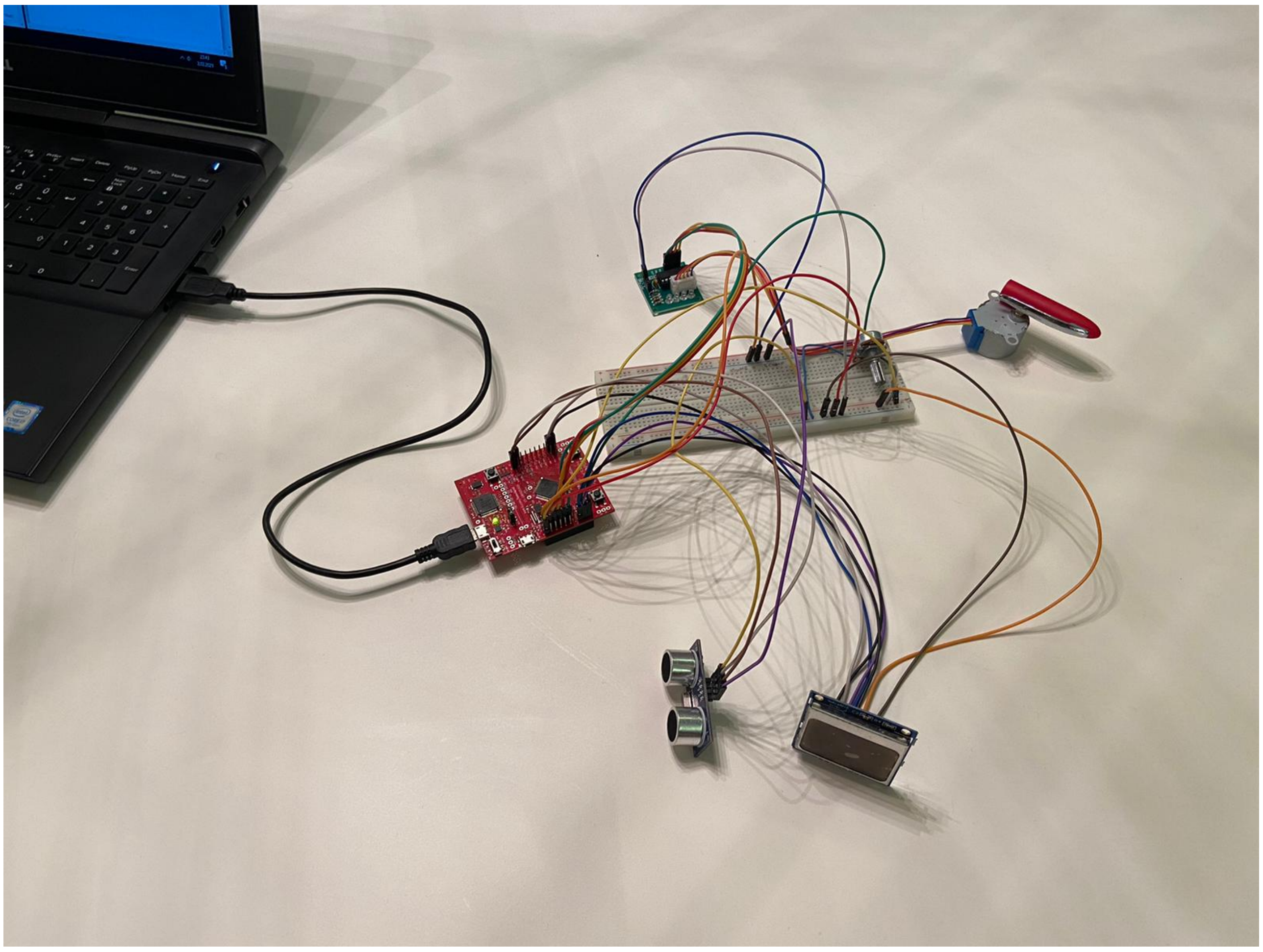

Figure 1: Setup

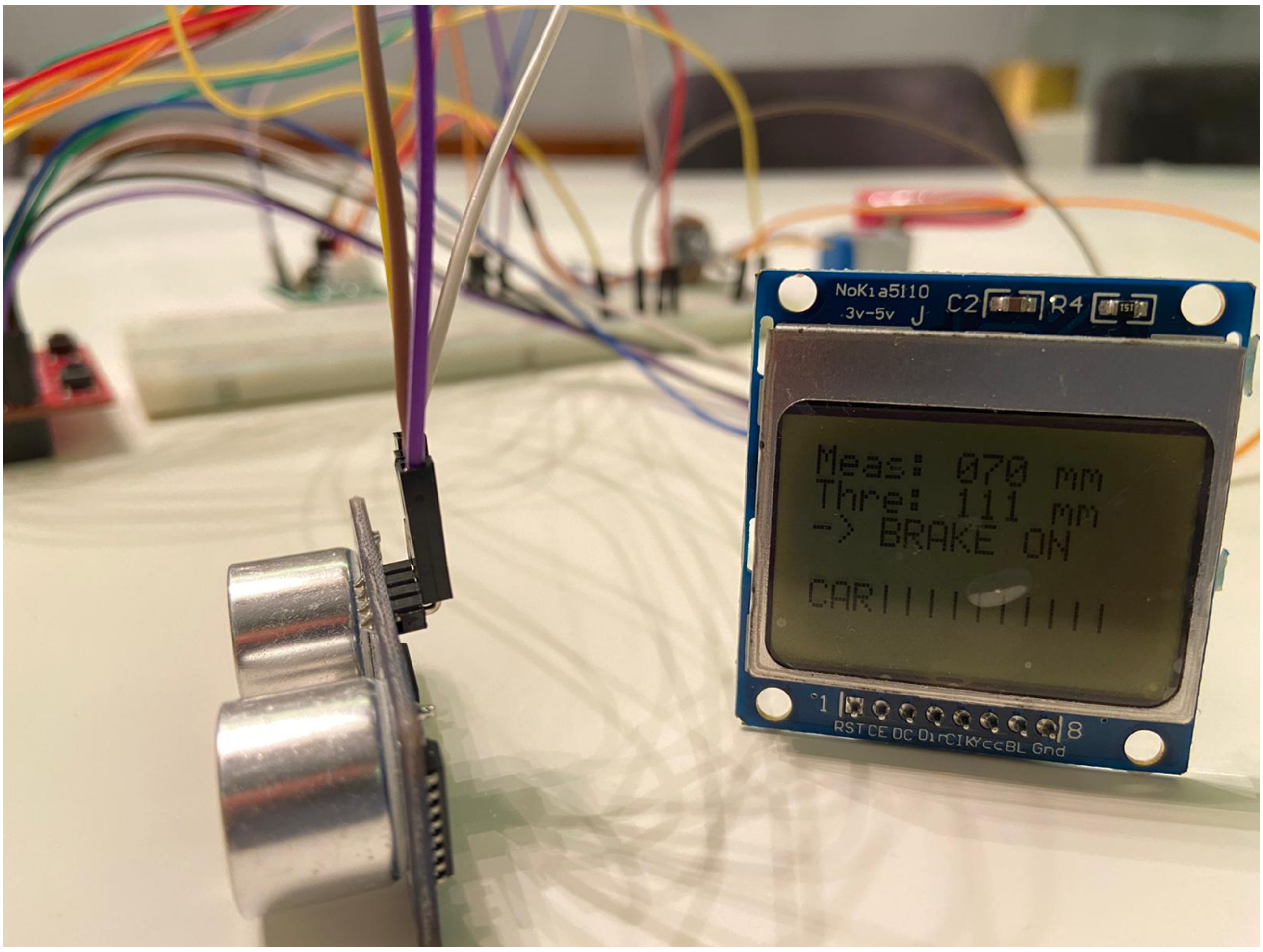

Figure 2: Braking Mode

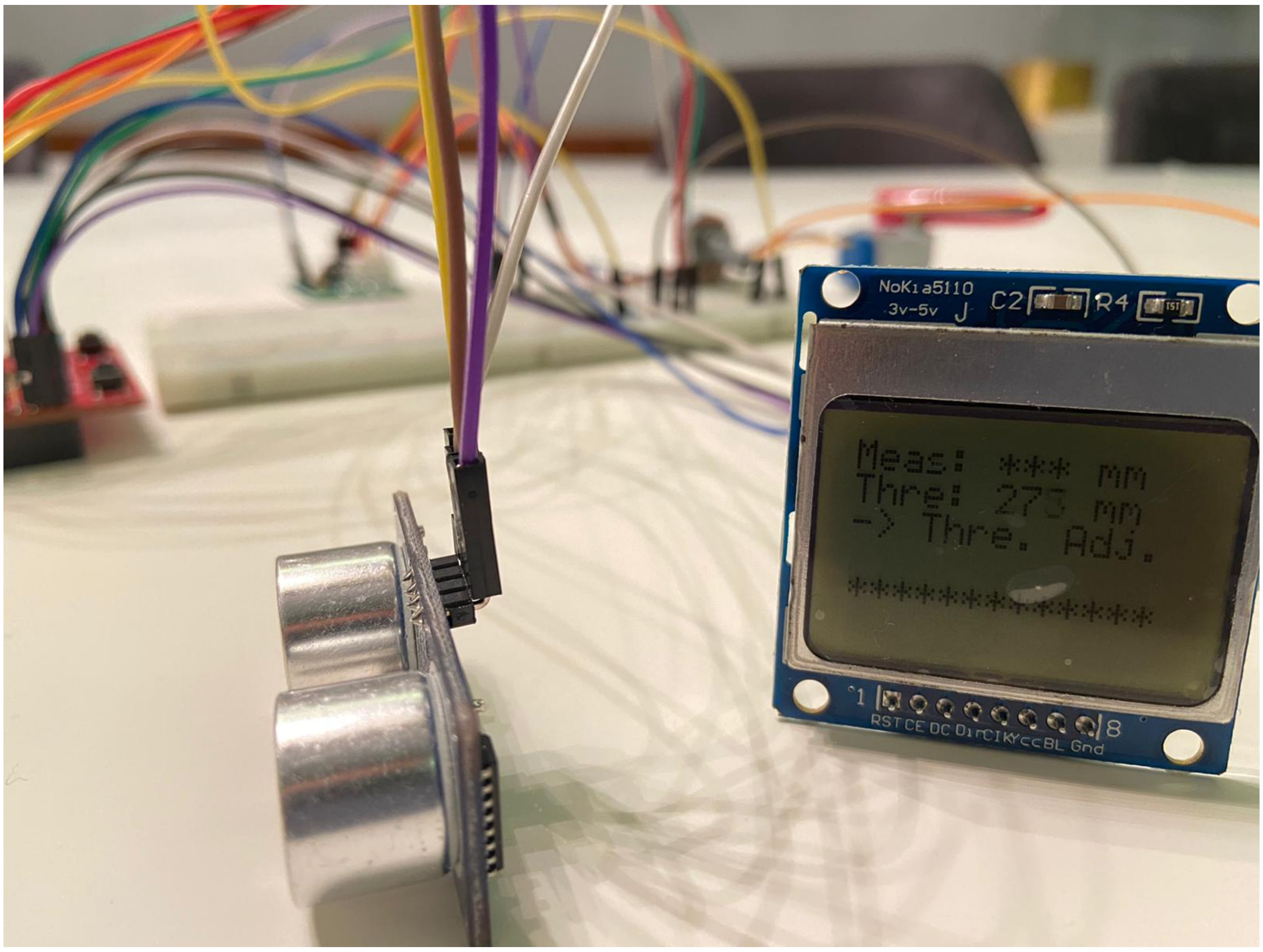

Figure 3: Threshold Adjustment Mode

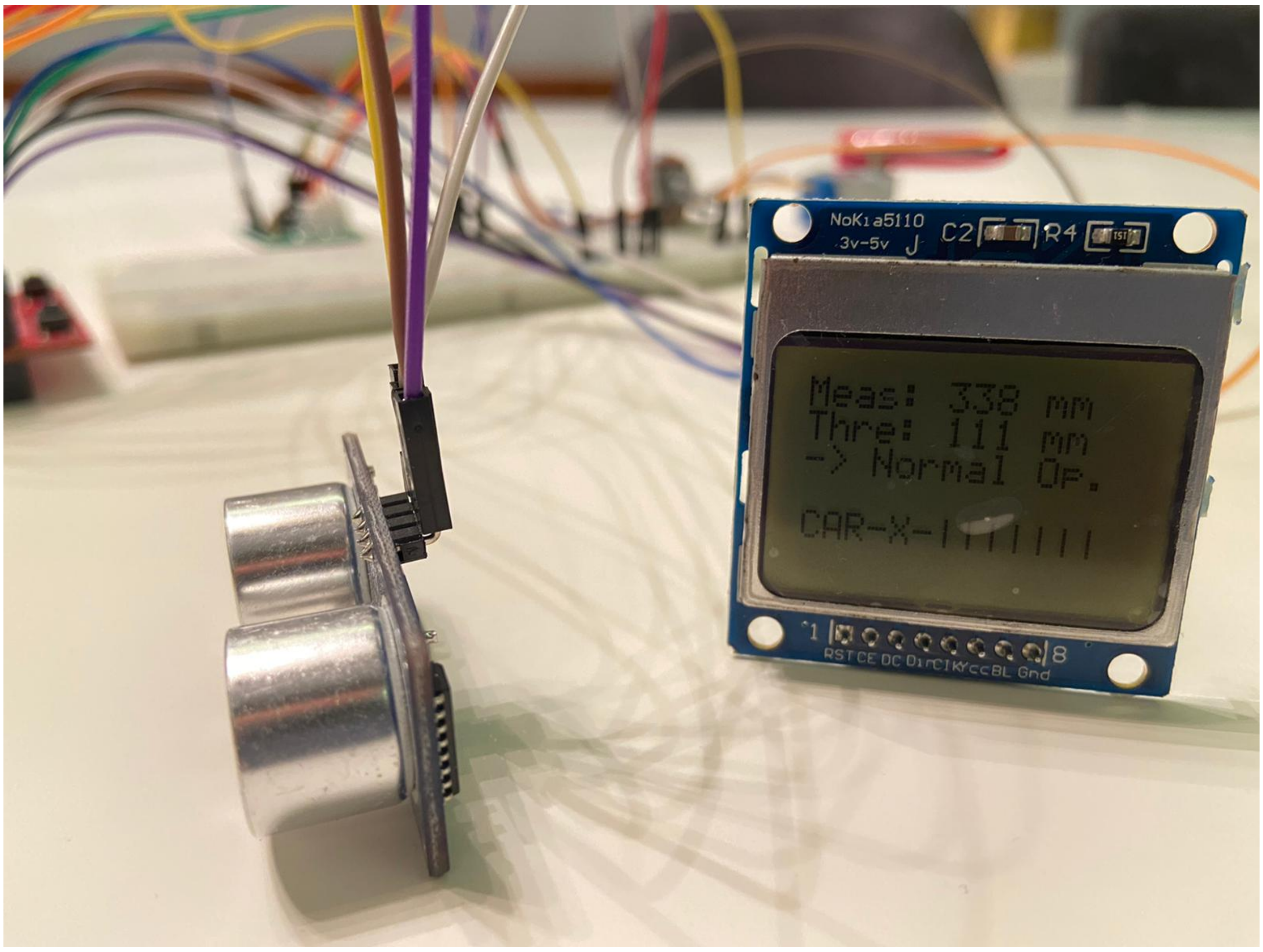

Figure 4: Normal Operation Mode

2 Module Explanation

2.1 Nokia LCD Module

Pins of Nokia LCD is connected to Tiva Board as follows:

```
Vcc (3.3V) --> 3.3V
GND --> GND
CE (SSI0-Fss) --> PA3
RST (Reset) --> PA7
DC (Data/Command) --> PA6
DN (SSI0-Tx) --> PA5
CLK (SSI0-Clk) --> PA2
```

2.1.1 Initialization Routine

Firstly, GPIO initializations are made. PortA clock is enabled. Each pin is configured in terms of directions, digital or analog, having alternate function.
Then, SSI0 clock is opened. For 2 MHz baud rate configuration; PIOSC is set to 16 MHz, Prescaler divisor(CPSDVSR) is set to 4 and SCR is set to 1. SPH and SPO bits are cleared, Freescale frame format is chosen and 8-bit data format is chosen. After that, SSI is enabled. Then, LCD memory is reset through writing to RST pin of LCD. Then, LCD setup begins by setting DC pin of LCD low. PD=0, V=0, H=1 is selected for SSI communication and Send 1Byte subroutine is used, which is explained afterwards. Contrast is set. Temperature Coefficient is set. Bias is set. Control mode is set to normal mode. After all these settings are done, wait until the SSI is done by checking SSI status register.

2.1.2 Send 1Byte Routine

SSI protocol is used for serial communication. In order to send 1 Byte information to LCD, we first look at SSI0 status register for buffer. After buffer is done, we can send our data through storing to SSI0 data register.

2.1.3 Clear Screen Routine

PA6 pin is set low for Command. H=0 is set and X Y cursor coordinates of LCD is set to zero using send 1Byte routine. Status register is checked after settings until SSI is done. PA6 pin is set high for data writing. Then, each pixel (84*48=504 pixel) is cleared by writing zero (blank ascii character is represented as zero value) to every pixel. Status register is checked after settings, until SSI is done.

2.1.4 Print Char Routine

PA6 pin is set high for data writing. Predefined ASCII table adress is loaded to a register and from that register we can find the ASCII print value of a character. Each ASCII print value is represented in 5 pixel width and 8 pixel height. ASCII table is defined in the beginning of the code. 1 empty column (which is zero value; blank character) is added after printing each char to separate the characters. Values are sent through send 1Byte routine and status register is checked after sending.

2.1.5 Print String Routine

Print string subroutine is simply done by printing the characters in a string that is stored in an adress. Print char subroutine is used until the end of transmission character (has an ASCII value of 4) is reached.

2.1.6 Set X-Y Coordinates Routine

In order to print anywhere we like, we need to set the cursor coordinates beforehand. For setting the cursor coordinates, firstly PA6 pin is set low for Command writing. H=0 is set. X coordinate (0-83 value) and Y coordinate (0-5 value) that is passed through registers are set using send 1Byte routine.

2.1.7 Print 3-digit Number Routine

3 digit number is passed via register. If the value is greater than 999, register value is made 999. With dividing and multiplying, the hex value is converted to decimal value. For each digit Print Char Nokia routine is called.

2.1.8 Print Progress Bar Routine

Measurement and threshold value are passed via registers. Progress bar coordinates are set. 3rd digits of measurement and threshold value is found and pushed to stack to be used afterwards. Applying different arithmetic operations, "-", "X", "—" characters are printed. Note that this subroutine is branched to braking mode and normal mode.

2.1.9 Normal Mode UI Routine

This subroutine will be called in main code. Firstly, screen is cleared using clear screen routine. X-Y coordinates are set to zero, which is the upper left corner of LCD and "Meas: mm" message is printed. Y coordinates are set to 1, which is row 1 and "Thre: mm" message is printed. Then, "Normal op." and "CAR" messages are printed in a similar manner. Then, measurement and threshold values that are passed through registers are printed via Print 3-digit Number Routine. Progress bar is printed via Print Progress Bar Routine.

2.1.10 Adjustment Mode UI Routine

This subroutine will be called in main code. It will be enabled or disabled by SW1 and its threshold value can be changed via potentiometer in main code. "Meas: mm", "Thre: mm", "Thre. Adj." and "*******" messages are printed in same way as in normal mode UI routine. Then, threshold values that are passed through a register is printed via Print 3-digit Number Routine.

2.1.11 Braking Mode UI Routine

This subroutine will be called in main code. It will be enabled or disabled by SW2. "Meas: mm", "Thre: mm", "Brake ON." and "CAR" messages are printed in same way as in normal mode UI routine. Threshold and measurement values that are passed through a register is printed via Print 3-digit Number Routine. Progress bar is printed via Print Progress Bar Routine.

2.2 ADC Module:

Potentiometer is used for adjusting the analog value and ADC module converts it to digital by sampling. ADC0 and PortE pin3 is used. PortE pin3 is connected to middle pin of potentiometer whereas other pins are connected to 3.3V and ground.

2.2.1 ADC initialization:

PortE clock is enabled. ADC0 clock is enabled. PE3 is configured as input, alternate funcion and analog. Software triggering is configured. Sample Sequencer 3 is disabled for configuration. 125k/samples is configured. Sample Sequencer 3 is enabled after configuration.
We will use SW1 and SW2 in main code. SW1 is PF4 and SW2 is PF0. PortF GPIO clocks are enabled and lock register is disabled for PortF. Directions are set to digital input and pull-up registers are enabled for switches.

2.2.2 ADC sampling:

Sampling is initiated by enabling SS3 using PSSI register. Sample complete is checked by RIS register. Value is stored in SSFIFO3 register. Max value of this register is 4096 and max value we want is 999. Knowing that ADC value is proportioned to value between 0 and 999. After conversion interrupt is cleared by writing to ISC register and sampling is finished by disabling PSSI register.

2.3 HC-SR04 Module:

HC-SR04 distance sensor has VCC, GND, ECHO and TRIG pins. Distance sensing is done this way. TRIG pin is triggered by a short pulse (approx. greater than 10us) and sends a short ultrasonic sound via its first speaker. Then, ECHO pin gathers this signal and converts it to digital value.

2.3.1 Trigger Routines:

Timer0A will be used for periodic pulse. It is designed to be HIGH for 15us and LOW for 100ms.

Pulse Initialization:
PortF GPIO clock is enabled. PF2 is made digital output. Timer0 clock is enabled. Timer disabled for configuration. Timer configured to 16-bit, periodic, count-down mode. Clock is divided to be 1us. Timeout interrupt is enabled. Timer interrupt priorities is set and interrupt is enabled.

Timer0A Handler:
When interrupt is called, this interrupt handler looks for PF2 data. If pin is low, it

makes high and changes interval load register value to 100000, which represents 100ms. If pin is high, it makes low and changes interval load register value to 15, which represents 15us.

2.3.2 Echo Routines:

Timer3A will be used for detecting pulse, more precisely detecting edges. Timer3A is initialized first, then edge detection routine will be called in main code.

Timer3A Initialization:
Since signal edge is detected by a pin, gpio port initialization needs to be done. PortB clock is enabled. PortB pin2 is set to digital input, alternate function and set to timer3. Timer3 clock is enabled. Timer3 is disabled for configuration. Edge time, capture, count-up mode is set. Clock is divided to 1us using prescaler. Both edge detection mode is set. Interval load register is loaded. Timer interrupt priority is set and interrupt is enabled. Timer3a is enabled.

Echo Detection Routine:
This routine (named as ultrasound) will be called in main code. Firstly, RIS register is checked whether an edge is detected or not. When an edge is detected, interrupt flag is cleared. Time at edge detection is captured. Period and pulse width is found by arithmetic operations. In order to be more accurate, this process is done for 10 cycles and more precise results have been got. After the overall period and pulse width is found, we apply the distance and time formula for ultrasonic sensor. Then the value is passed via registers to be used afterwards.

2.4 Stepper Motor Module:

I couldn't manage to operate stepper motor with GPTM interrupts.

2.5 Main Code:

All subroutines to be called in the main code, is exported in main code area. Pulse initialization and timer3 configuration routines are called for ultrasonic sensor initialization. Nokia initialization routine is called. ADC initialization routine is called. initial threshold value is set. Then loop begins.
Measurement value is taken from echo detection routine (ultrasound). Normal mode subroutine is called, which includes normal mode UI subroutine only. If measurement value is less than threshold value, braking mode is called. In braking mode measurement value is get and braking mode UI routine is called. After a small delay, switch 2 is polled. I couldn't use interrupts for switches. After releasing switch 2, braking mode subroutine ends. Then, switch 1 is checked with debouncing. If switch 1 is pressed, then after the release of switch

1, adjustment subroutine is called. In adjustment subroutine, ADC sampling routine is called and with the threshold value, adjustment mode UI routine is called. Again, switch 1 is checked and if it is pressed released, adjustment mode routine ends.