Experiment No 8
Analog to Digital Convertor (ADC) Interface with STM32

Objectives:

- To learn the basics of ADC and its interface with STM32F103 Board
- Interfacing Analogue Sensors with STM32F103 by using its ADC Channels.
- Displaying the output of ADC on a 16x2 LCD display.

8.1 ADC Introduction:

The 12-bit ADC is a successive approximation analog-to-digital converter. It has up to 18 multiplexed channels allowing it measure signals from sixteen external and two internal sources. A/D conversion of the various channels can be performed in single, continuous, scan or discontinuous mode. The result of the ADC is stored in a left- aligned or right-aligned 16-bit data register. The analog watchdog feature allows the application to detect if the input voltage goes outside the user-defined high or low thresholds. The ADC input clock is generated from the PCLK2 clock divided by a prescaler and it must not exceed 14 MHz.

8.2 ADC main features:

- 12-bit resolution
- Interrupt generation at End of Conversion, End of Injected conversion and Analog watchdog event
- Single and continuous conversion modes
- Scan mode for automatic conversion of channel 0 to channel „n“
- Self-calibration
- Data alignment with in-built data coherency
- Channel by channel programmable sampling time
- External trigger option for both regular and injected conversion
- Discontinuous mode
- Dual mode (on devices with 2 ADCs or more)

ADC conversion time:

- STM32F103xx performance line devices: 1 µs at 56 MHz (1.17 µs at 72 MHz)
- STM32F101xx access line devices: 1 µs at 28 MHz (1.55 µs at 36 MHz)
- STM32F102xx USB access line devices: 1.2 µs at 48 MHz
- STM32F105xx and STM32F107xx devices: 1 µs at 56 MHz (1.17 µs at 72MHz)
- ADC supply requirement: 2.4 V to 3.6 V
- ADC input range: VREF- ≤ VIN ≤ VREF+
- DMA request generation during regular channel conversionThe block diagram of the

ADC is shown below:

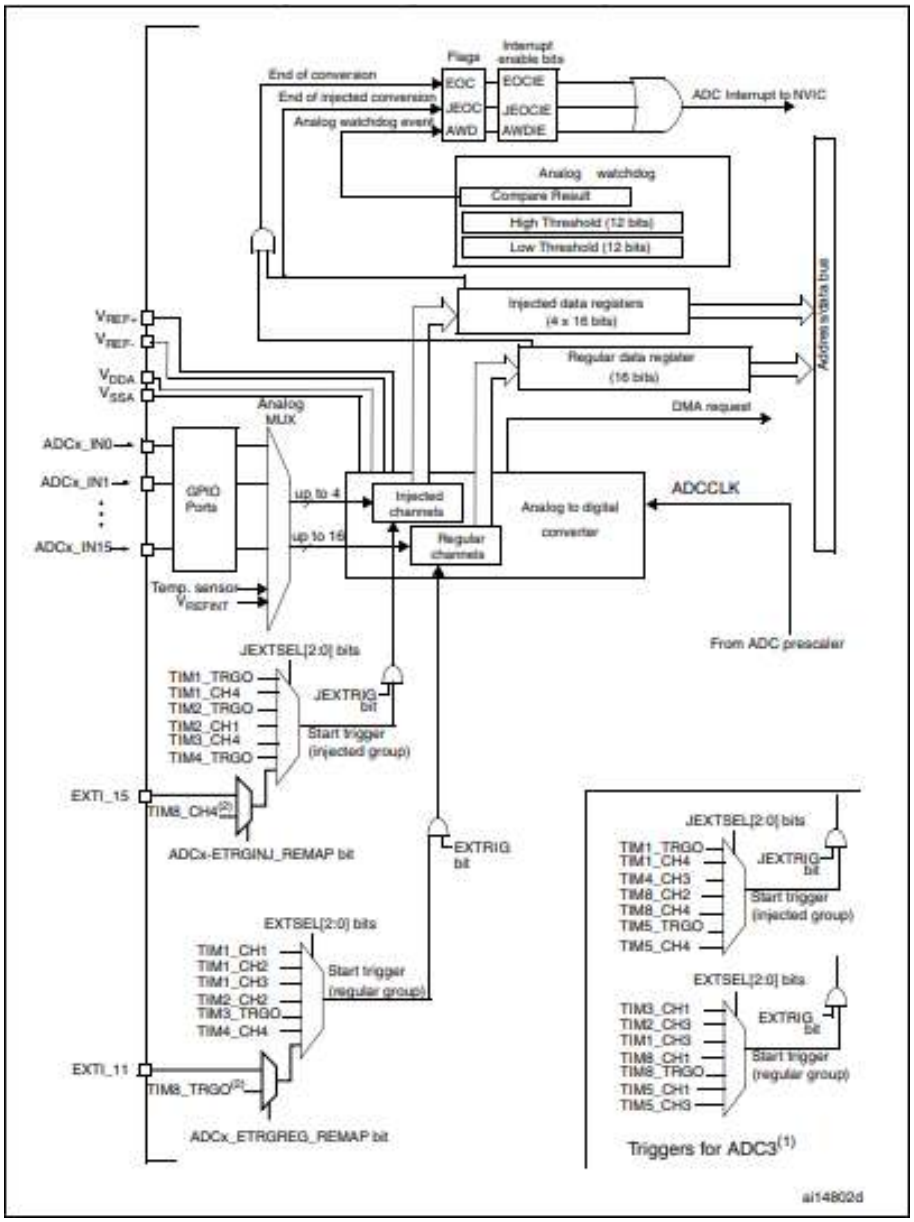


Figure 1: Single ADC Block Diagram

Table below gives the ADC pin description.

Name	Signal type	Remarks
V_{REF+}	Input, analog reference positive	The higher/positive reference voltage for the ADC, 2.4 V ≤V_{REF+} ≤V_{DDA}
V_{DDA}(1)	Input, analog supply	Analog power supply equal to V_{DD} and 2.4 V ≤V_{DDA} ≤3.6 V
V_{REF-}	Input, analog reference negative	The lower/negative reference voltage for the ADC, V_{REF-} = V_{SSA}
V_{SSA}(1)	Input, analog supply ground	Ground for analog power supply equal to V_{SS}
ADCx_IN[15:0]	Analog signals	Up to 21 analog channels(2)

1. V_{DDA} and V_{SSA} have to be connected to V_{DD} and V_{SS}, respectively.
2. For full details about the ADC I/O pins, refer to the "Pinouts and pin descriptions" section of the corresponding device datasheet.

8.3 16x2 LCD:

Liquid Crystal Display (LCD) is very commonly used electronic display module and having a wide range of applications such as calculators, laptops, mobile phones etc. 16×2 character LCD display is very basic module which is commonly used in electronics devices and projects. 16×2 LCD is named so because; it has 16 Columns and 2 Rows. There are a lot of combinations available like, 8×1, 8×2, 10×2, 16×1, etc. But the most used one is the 16×2 LCD, hence we are using it here. It can display 2 lines of 16 characters. Each character is displayed using 5×7 or 5×10 pixel matrix. Interfacing 16×2 LCD with STM32F103 using Keil uVision5 is a bit complex because there are no powerful libraries in Keil. To solve this problem we will be using an open source LCD library which includes commonly used features, you just need to include therequired files and use them.

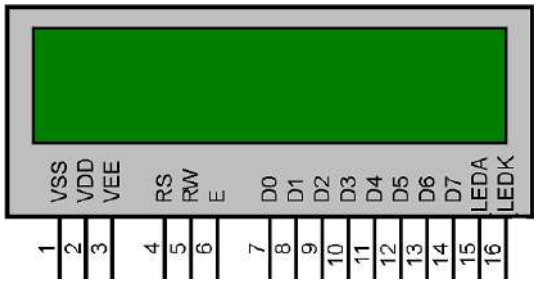


Figure-2: 16x2 LCD Schematic



Figure-3: 16x2 LCD

The pin configuration of 16x2 LCD is shown below:

NAME	FUNCTION
VSS	Ground Pin
VDD	+5V Input Pin
VEE	Contrast Adjust Pin
RS	Register Select
R/W	Read/Write Pin
E	Enable Pin
D0-D7	Data Pins (8 Pins)
LED A	Anode Pin (+5V)
LED K	Cathode Pin (GND)

The LCD can work in two different modes, namely the 4-bit mode and the 8-bit mode. In 4-Bit mode we send the data nibble by nibble, first upper nibble and then lower nibble. For those of you who don‟t know what a nibble is: a nibble is a group of four bits, so the lower four bits (D0- D3) of a byte form the lower nibble while the upper four bits (D4-D7) of a byte form the higher nibble. This enables us to send 8- Bit data.

Whereas in 8-Bit mode we can send the 8-bit data directly in one stroke since we use all the 8 data lines.

Here in this project we will use the most commonly used mode which is 4-Bit mode. In 4-Bit mode we can save 4 pins of microcontroller and also reduce the wiring overhead.

16x2 also uses HEX code to take any command, there are many hex commands for LCD like to move the cursor, select the mode, shift the control to second line etc.

8.4 Required Configurations for ADC:

```
void ADC1_Init()
{
	// Initialization struct
	ADC_InitTypeDef ADC_InitStruct;
	GPIO_InitTypeDef GPIO_InitStruct;

	// Step 1: Initialize ADC1
	RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC1, ENABLE);

	ADC_InitStruct.ADC_ContinuousConvMode = DISABLE;
	ADC_InitStruct.ADC_DataAlign = ADC_DataAlign_Right;
	ADC_InitStruct.ADC_ExternalTrigConv = DISABLE;
	ADC_InitStruct.ADC_ExternalTrigConv = ADC_ExternalTrigConv_None;
	ADC_InitStruct.ADC_Mode = ADC_Mode_Independent;
	ADC_InitStruct.ADC_NbrOfChannel = 1;
	ADC_InitStruct.ADC_ScanConvMode = DISABLE;
	ADC_Init(ADC1, &ADC_InitStruct);
	ADC_Cmd(ADC1, ENABLE);
	// Select input channel for ADC1
	// ADC1 channel 0 (PA0)
	ADC_RegularChannelConfig(ADC1, ADC_Channel_0, 1, ADC_SampleTime_55Cycles5);

	// Step 2: Initialize GPIOA (PA0)
	RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
	GPIO_InitStruct.GPIO_Pin = GPIO_Pin_0;
	GPIO_InitStruct.GPIO_Mode = GPIO_Mode_AIN;
	GPIO_Init(GPIOA, &GPIO_InitStruct);
}
```

8.5 Required GPIO Configurations for LCD:

ADC_Working.c | lcd16x2.h

```
23  /** Port and pin definition for 4-bit mode ------------------------
24  // LCD control lines (must be on the same port)
25  #define LCD16X2_RCC_GPIO_CONTROL  RCC_APB2Periph_GPIOB
26  #define LCD16X2_GPIO_CONTROL      GPIOB
27  #define LCD16X2_GPIO_RS           GPIOB
28  #define LCD16X2_GPIO_RW           GPIOB
29  #define LCD16X2_GPIO_EN           GPIOB
30  // LCD data lines (must be on the same port)
31  #define LCD16X2_RCC_GPIO_DATA     RCC_APB2Periph_GPIOA
32  #define LCD16X2_GPIO_DATA         GPIOA
33  #define LCD16X2_GPIO_D4           GPIOA
34  #define LCD16X2_GPIO_D5           GPIOA
35  #define LCD16X2_GPIO_D6           GPIOA
36  #define LCD16X2_GPIO_D7           GPIOA
37  // Pin definition
38  #define LCD16X2_PIN_RS  GPIO_Pin_13
39  #define LCD16X2_PIN_RW  GPIO_Pin_14
40  #define LCD16X2_PIN_EN  GPIO_Pin_15
41  #define LCD16X2_PIN_D4  GPIO_Pin_1    // 4-bit mode LSB
42  #define LCD16X2_PIN_D5  GPIO_Pin_2
43  #define LCD16X2_PIN_D6  GPIO_Pin_3
44  #define LCD16X2_PIN_D7  GPIO_Pin_4    // 4-bit mode MSB
45
```

8.6 Libraries to include:

```
1  #include "stm32f10x.h"
2  #include "stm32f10x_rcc.h"
3  #include "stm32f10x_gpio.h"
4  #include "stm32f10x_adc.h"
5  #include "delay.h"
6  #include "lcd16x2.h"
7  #include <stdio.h>
8
9  void ADC1_Init(void);
10 uint16_t ADC1_Read(void);
11
12 uint16_t adcValue;
13 char sAdcValue[5];
14
```

8.7 Main Function:

```
int main(void)
{
  float voltage;

  DelayInit();
  lcd16x2_init(LCD16X2_DISPLAY_ON_CURSOR_OFF_BLINK_OFF);

  lcd16x2_puts("VOLTAGE = ");
  lcd16x2_gotoxy(3,2);

  DelayMs(10);

  // Initialize ADC
  ADC1_Init();

  while (1)
  {
    // Read ADC value
    adcValue = ADC1_Read();

    // Convert ADC value to Voltage using formulae
    voltage = (float)adcValue/4095 * 3.3;

    // Convert Voltage(float) into Voltage(string)
    sprintf(sAdcValue, "%f v", voltage);

    // Display ADC value to LCD
    lcd16x2_gotoxy(3,2);
    lcd16x2_puts(sAdcValue);

    DelayMs(5);
  }
}
```

8.8 ADC Read Function:

```
uint16_t ADC1_Read()
{
  // Start ADC conversion
  ADC_SoftwareStartConvCmd(ADC1, ENABLE);
  // Wait until ADC conversion finished
  while (!ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC));

  return ADC_GetConversionValue(ADC1);
}
```

8.9 Programming in KEIL:

To program the interfaces in Keil, the first step is to create a new uVision Project and selectingthe device which is STM32F103C8 just like we already did in the previous labs.

After saving the project in your desired directory, the device will be selected in the „SelectDevice for Target" dialogue box.

After that, include the libraries as shown in the figure below:

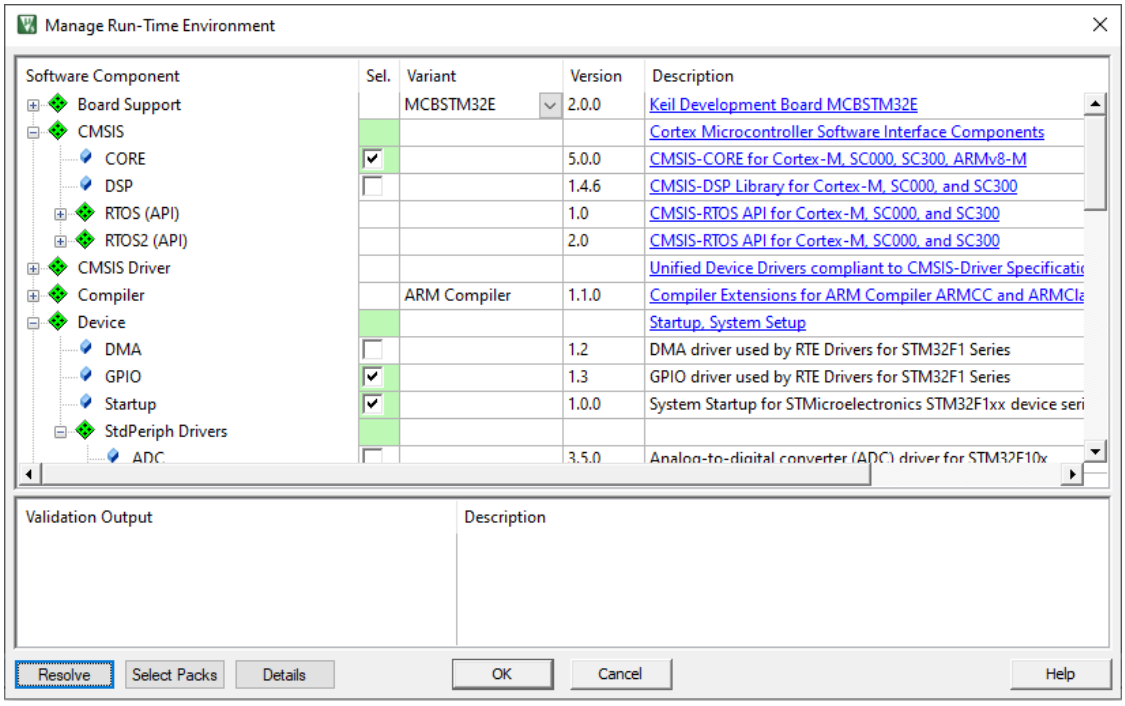


Figure 2: Adding required libraries

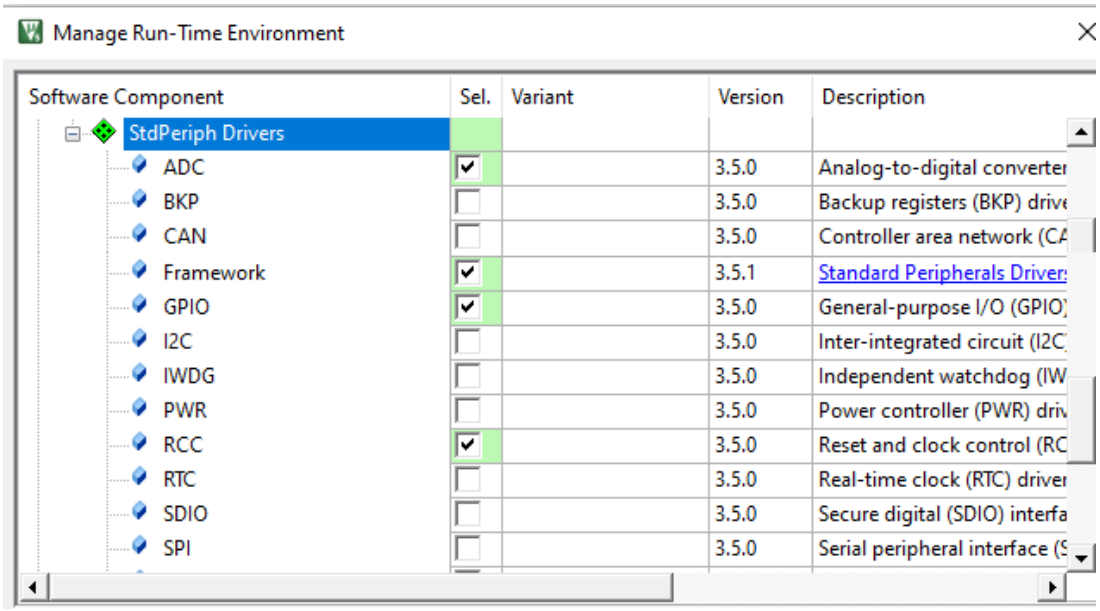


Figure 3: Adding required libraries

Then add a new item into your source group. Select the “C File (.c)” option, name your C fileand save it in your desired directory.

After adding the new C file add the required header files in the Source Group as shown below.

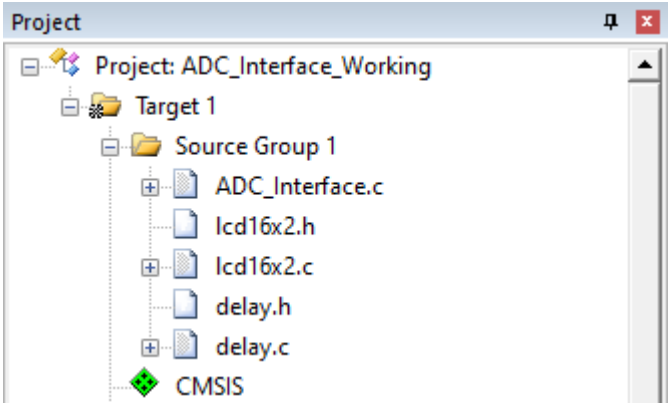


Figure 6: Adding required header files

Lab Task (Psychomotor)

Activity 01:

Write a C-Program that can take an analogue input from a potentiometer, the program will convert the analogue input into a digital output using the ADC of STM32F103 board. Program will convert the digital output into a string variable and displays the voltages on the LCD.

Activity 02:

Write C-Program which will convert the input analogue voltage into a digital output using the ADC of STM32F103 board. Program will write "Temperature =" in the first row of the LCD and will write the digital output (Temperature) converted by ADC in the second row. ADC will be getting the analogue input from a Temperature sensor. Implement the design using STM32F103 board and show the results to lab instructor.

Lab Exercise and Summary

Summary should cover Introduction, Procedure, Data Analysis and Evaluation.

Student's Signature: _____ Date: _____

Lab Task (Cognitive)

Question 1:

The 12-bit ADC of the microcontroller uses VREF+ = 3.3 V and VREF- = 0 V. Use the relation: ADC value = (VIN / VREF) x 4096.

a) Determine the number of quantization levels and the voltage step size (in mV) of the ADC. **[3 Marks]**

b) Determine the ADC output (decimal and hexadecimal) for an input of 1.65 V. **[3 Marks]**

c) An LM35 temperature sensor (10 mV/°C) is connected to the ADC and the conversion result is 372. Determine the input voltage and the temperature. **[4 Marks]**

LABORATORY SKILLS ASSESSMENT (Cognitive)

Obtained Marks	**/10**

Instructor's Signature: _____ Date: _____

LABORATORY SKILLS ASSESSMENT (Psychomotor)

Psychomotor Activity 1: Experimental Setup

S. No	Criteria	Performance	Description	Marks
1	**ADC Hardware & Sensor Setup (4)**	Level 4	Correctly prepares the required ADC hardware and sensor setup for the experiment independently and according to the laboratory requirements.	4
		Level 3	Correctly prepares the required ADC hardware and sensor setup with minor errors or assistance.	3
		Level 2	Partially prepares the required ADC hardware and sensor setup and requires considerable assistance.	2
		Level 1	Shows limited ability to prepare the required ADC hardware and sensor setup and requires substantial assistance.	1
2	**ADC Channel & Clock Configuration (4)**	Level 4	Correctly prepares the required ADC channel and clock configuration for the experiment independently and according to the laboratory requirements.	4
		Level 3	Correctly prepares the required ADC channel and clock configuration with minor errors or assistance.	3
		Level 2	Partially prepares the required ADC channel and clock configuration and requires considerable assistance.	2
		Level 1	Shows limited ability to prepare the required ADC channel and clock configuration and requires substantial assistance.	1
3	**Conversion Parameters (2)**	Level 2	Correctly sets the required conversion parameters for the experiment.	2
		Level 1	Sets the basic conversion parameters with minor errors or assistance.	1
Total Obtained				**/10**

If a student is absent:

Criteria	Description	Marks
Absent	Student is absent from the laboratory activity.	**0**

Psychomotor Activity 2: Implementation + Results

S. No	Criteria	Performance	Description	Marks
1	**ADC C Program (4)**	Level 4	Correctly implements the assigned ADC C program independently while following appropriate laboratory practices.	4
		Level 3	Implements the assigned ADC C program with minor errors or assistance and generally follows proper laboratory practices.	3
		Level 2	Partially implements the assigned ADC C program and requires considerable assistance or repeated guidance.	2
		Level 1	Implementation of the assigned ADC C program is largely incomplete, requires substantial assistance, or shows poor adherence to proper laboratory and safety practices.	1
2	**Execution & Debugging (2)**	Level 2	Correctly performs the required execution and debugging and verifies the expected operation.	2
		Level 1	Performs the basic execution and debugging with minor errors or assistance.	1
3	**ADC Reading & Display Analysis (4)**	Level 4	Obtains correct ADC reading and display analysis and appropriately verifies and interprets the required experimental results.	4
		Level 3	Obtains mostly correct ADC reading and display analysis with minor errors in verification or interpretation.	3
		Level 2	Obtains partially correct ADC reading and display analysis with noticeable errors or omissions.	2
		Level 1	ADC reading and display analysis is largely incorrect or incomplete and requires substantial assistance.	1
Total Obtained				**/10**

If a student is absent:

Criteria	Description	Marks
Absent	Student is absent from the laboratory activity.	**0**

Instructor's Signature: _____ Date: _____

LABORATORY SKILLS ASSESSMENT (Affective)

Affective Activity: Lab Report + Viva Voce

S. No	Criteria	Performance	Description	Marks
1	**Lab Report Completeness & Organization (4)**	Level 4	Report is complete, well-organized, properly presented, and demonstrates responsible completion of the assigned laboratory work.	4
		Level 3	Report is complete with minor omissions or organizational issues and demonstrates satisfactory responsibility for the assigned work.	3
		Level 2	Report is partially complete with several omissions and demonstrates limited responsibility for the assigned work.	2
		Level 1	Report has major omissions or poor organization and demonstrates minimal responsibility for the assigned work.	1
2	**ADC Output & Conclusion (2)**	Level 2	The ADC output and conclusion are correctly presented and discussed, demonstrating ownership of the assigned work.	2
		Level 1	The ADC output and conclusion are presented with minor errors or omissions.	1
3	**Viva Voce (4)**	Level 4	Clearly explains the assigned experiment, ADC resolution, channel configuration, program explanation, working, and results, demonstrating individual responsibility, participation, and effective communication.	4
		Level 3	Explains most of the assigned experiment and results correctly, with minor difficulty, and demonstrates good individual participation.	3
		Level 2	Explains basic aspects of the assigned experiment with some prompting and demonstrates basic participation.	2
		Level 1	Demonstrates limited understanding or involvement and requires substantial prompting to explain the assigned work.	1
Total Obtained				**/10**

The student has not submitted the report and has not appeared in the Viva Voce.

Criteria	Description	Marks
Lab Report and Results Not Submitted	Student has not submitted the report and results file.	**0**
Absent in Viva Voce	Student is absent from the viva voce.	**0**

Instructor's Signature: _____ Date: _____