Algorithm For the Measure 0 to 100Vdc Using PIC18F4580 Micro-Controller Code:

1. Include necessary header files:

- xc.h for microcontroller-specific definitions.

- stdio.h for standard input/output operations.

2. Configure the microcontroller:

- Disable the Watchdog Timer.

3. Define constants:

- ADC\_CHANNEL is set to 0, indicating analog input channel 0.

4. Declare a function ADC\_Init () to initialize the ADC module:

- Disable the ADC module.

- Configure AN0 (channel 0) as an analog input.

- Set the ADC result format to right justified.

- Set the acquisition time to 12 TAD.

- Set the conversion clock to FOSC/16.

- Enable the ADC module.

5. Declare a function ADC\_Read () to read the ADC value:

- Start the conversion.

- Wait for the conversion to complete.

- Return the 10-bit ADC result by combining ADRESH and ADRESL.

6. Declare a function Delay () for a delay:

- Use for loops to introduce a delay by iterating 1000 number of times.

7. Define the main () function:

- Declare variables adcValue (unsigned int) and voltage (float).

- Set all PORTA pins as inputs.

- Call ADC\_Init () to initialize the ADC module.

- Enter an infinite loop.

- Inside the loop:

- Read the ADC value using ADC\_Read () and store it in adcValue.

- Convert the ADC value to voltage by multiplying it by 5.0 and dividing by 1023.0 (assuming a 5V reference voltage).

- call a delay function for stability.

- Check if the voltage is within the desired range (between 0 and 100).

- If the voltage is within the range:

- Print the voltage value using printf function.

- The program continues to read and process the ADC value indefinitely.