**EMBEDDED SYSTEMS FISAC-2**

**LM35 Temperature Sensor on MBED Board**

**Team Details :**

**Team No: 1**

**Branch: INFORMATION TECHNOLOGY A1**

**Team Member Names:**

## **ANISH NOEL RASQUINHA - 200911002**

## **MOHAMMED AFNAN ALTHAF SAMAD - 200911006**

## **SHREYAS PRAVIN PHATAK - 200911005**

**Problem Statement :**

Interface a temperature sensor to NXP LPC1768 MBED board and calculate the temperature and display it on a remote device/LCD.

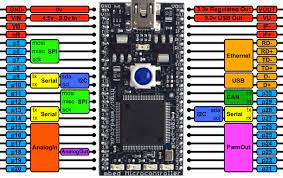
**Abstract :**

The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly proportional to the Centigrade temperature. It is a well-known, inexpensive temperature sensor, and because it is directly calibrated in degrees Celsius, the output voltage exactly correlates to readings in degrees Celsius. In comparison to linear temperature sensors calibrated in Kelvin, the LM35 device has an advantage because it does not require the user to deduct a significant constant voltage from the output in order to gain convenient Centigrade scaling. Its measuring range is from -55 to 150 degrees Celsius, with typical accuracy(ies) of 0.25 degrees Celsius at room temperature and 0.75 degrees Celsius over the entire range. The LM35 Temperature Sensor is offered in 4 distinct packages, including TO-CAN, TO-92, SOIC, and TO-220, and supports a broad variety of supply voltages from 4V to 30V.

**Hardware Used :**

The components used in the project include -

1. **MBED NXP LPC1768**: The mbed Microcontrollers are a series of ARM microcontroller development boards designed for rapid prototyping. The mbed NXP LPC1768 Microcontroller is designed for prototyping all sorts of devices, especially those including Ethernet, USB, and the flexibility of lots of peripheral interfaces and FLASH memory. It is packaged as a small DIP form-factor for prototyping with through-hole PCBs, stripboard, and breadboard. It includes a built-in USB FLASH programmer. The pinout diagram in Figure 12.1 shows the commonly used interfaces and their locations. Note that all the numbered pins (p5-p30) can also be used as DigitalIn and DigitalOut interfaces.



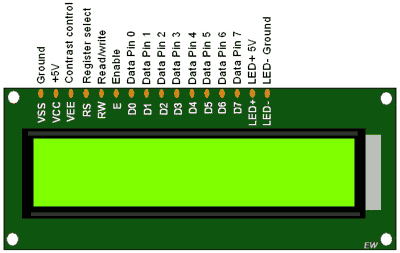
1. **LM35 Temperature Sensor**: LM35 is a temperature measuring device having an analog output voltage proportional to the temperature. It provides output voltage in Centigrade (Celsius). It does not require any external calibration circuitry. The sensitivity of LM35 is 10 mV/degree Celsius. As temperature increases, output voltage also increases. It is a 3-terminal sensor used to measure surrounding temperature ranging from -55 °C to 150 °C. LM35 gives temperature output which is more precise than thermistor output.

****

1. **Jumper Cables:** Female to Female jumpers were used for the experiment so as to connect the pins on the MBED NXP LPC1768 directly to the LM35.

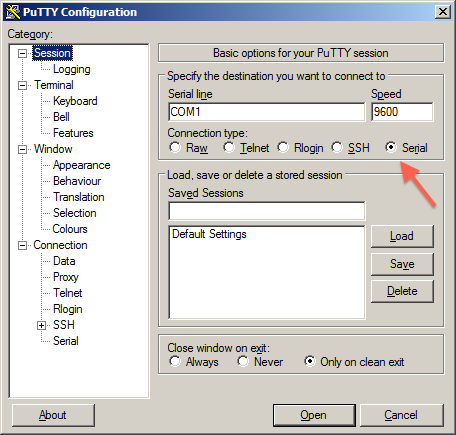


1. **PC:** To monitor the output of the MBEDcalculated Temperature readings.
2. **LCD:** LCDs (Liquid Crystal Displays) are used in embedded system applications for displaying various parameters and status of the system. LCD 16x2 is a 16-pin device that has 2 rows that can accommodate 16 characters each.

****

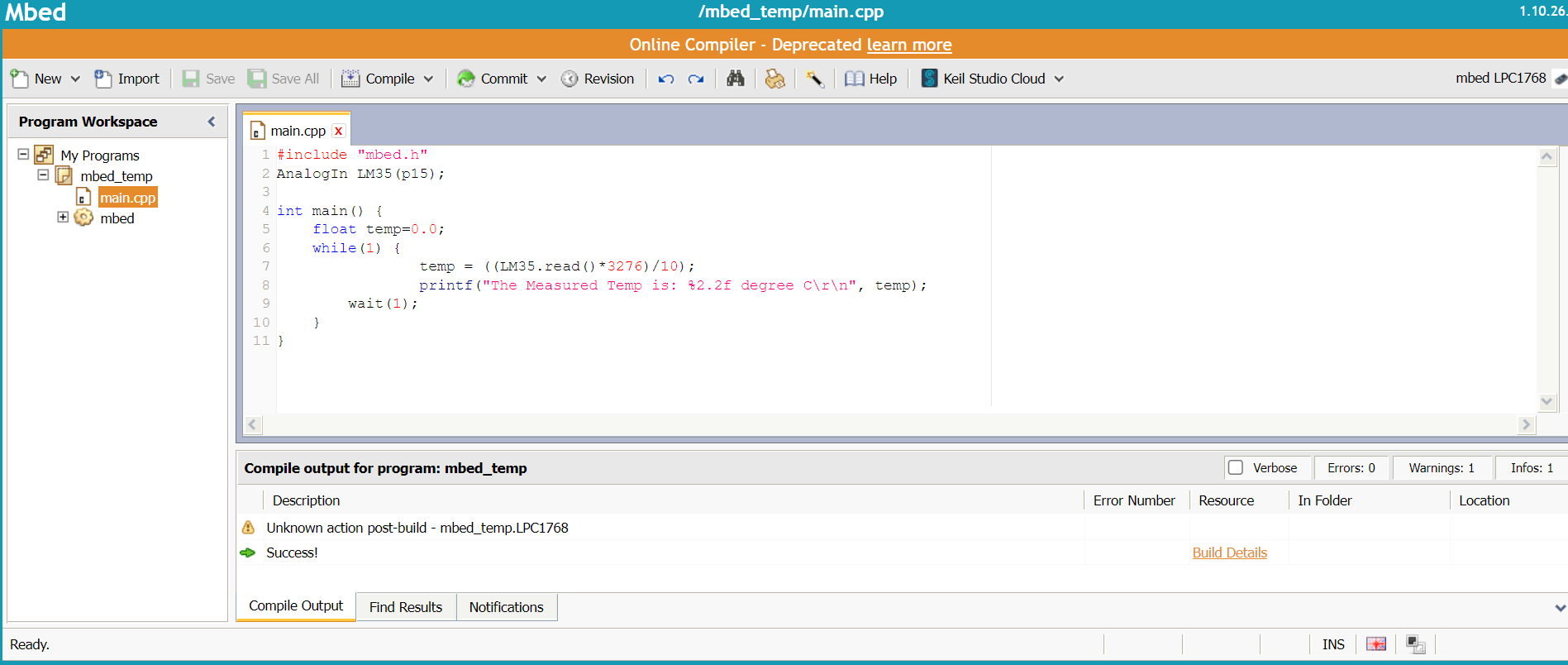
**Software Used :**

1. **PuTTY** : PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers.



1. **Mbed Online Compiler :** The Mbed Compiler provides a lightweight online C/C++ IDE that is pre-configured to let you quickly write programs, compile and download them to run on your Arm-based microcontroller. In fact, you don't have to install or set up anything to get running with Mbed. Because it is a web app, you can log in from anywhere and carry on where you left off, and you are free to work on Windows, Mac, iOS, Android, Linux, or all of them.

The compiler uses the professional Arm Compiler engine, so it produces efficient code that can be used free-of-charge, even in commercial applications. The IDE includes workspace version control, code formatting and auto generation of documentation for published libraries. The Mbed tools are focused on prototyping and are designed for fast experimentation, and complement other professional production-level tools; you can even export directly to other toolchains if you choose, as you progress to productise your design.



1. **Serial port drivers for Windows :** Drivers required for serial port configuration.
2. **Serial to Ethernet Connector:** Serial over Ethernet software allows you to share COM ports over the network and connect to remote serial devices like they were attached directly to your local computer. The program is designed to create virtual copies of real serial ports on remote PCs. If your computer lacks a serial port or you need additional COM interfaces for communicating with all of your COM-based devices, Serial over Ethernet software will let you create as many virtual serial ports as you need.

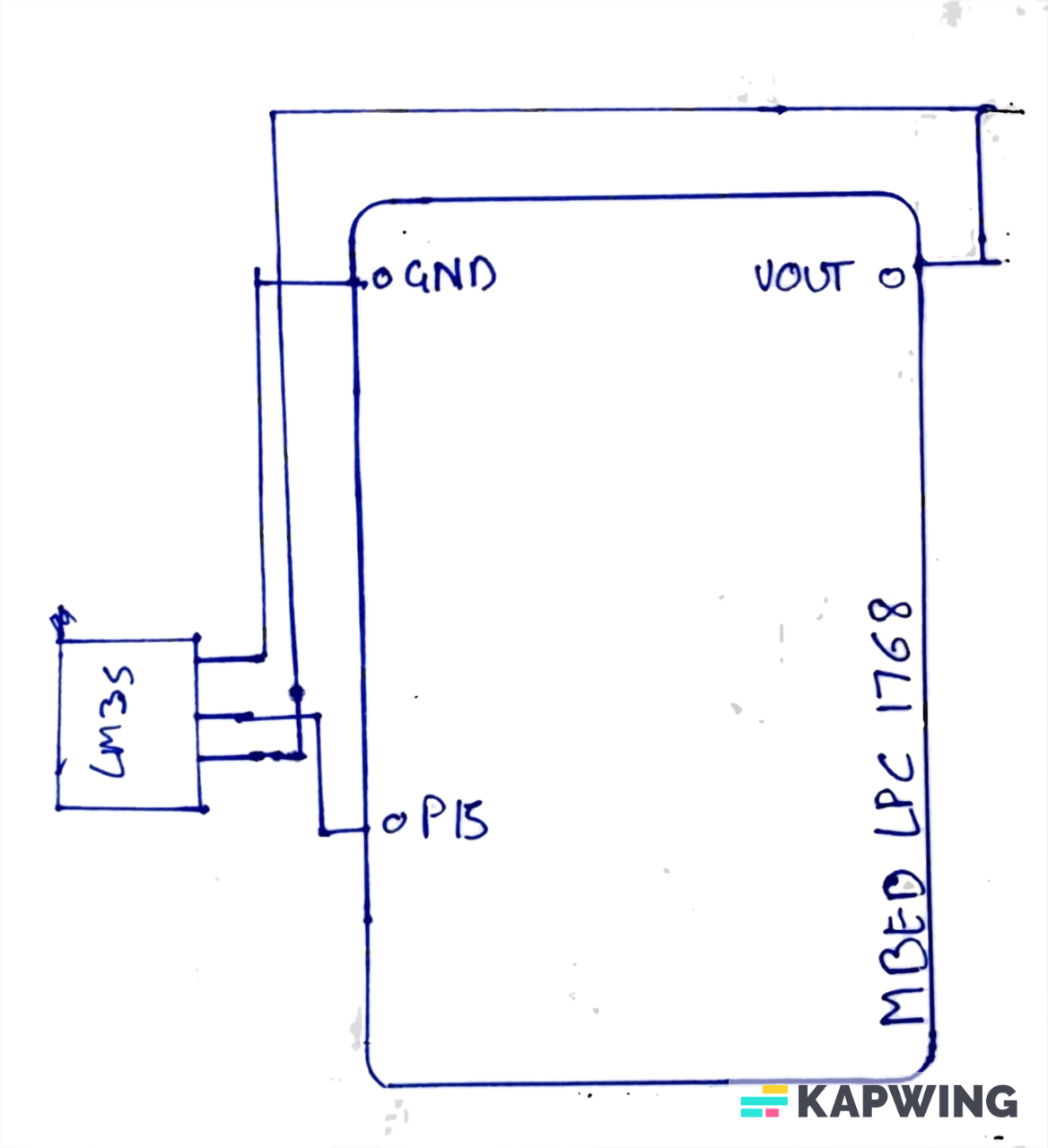


**Flow of Actions Performed to get the output:**

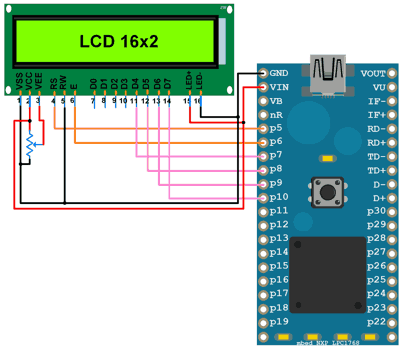
1. We interface the lm35 to the MBED NXP LPC1768 board using female to female. Connect the Gnd pin of the Mbed board to the top most pin of the LM35 sensor. We connect the middle pin of the LM35 to P15 on the MBED board. We connect the lowermost pin to the VOUT of the MBED board. The LCD display is also interfaced to the MBED board using P5 , P6 ,P7 P8, P9 and P10.
2. We connect the MBED board to the PC using a micro usb cable.
3. We write the code to read the temperature on the Online compiler, then compile it. A bin file is created.
4. We move the bin file into the MBEDs location, in the windows explorer.
5. Startup PuTTY, set the connection type to serial, and change the port number to 8(COM8), set baud rate to 9600.
6. Click on open to display a blank serial terminal window.
7. Click on the reset button to display the readings of the LM35 sensor on the LCD connected to the MBED board and also on the serial terminal window.

**Pin Diagram :**

**LM35 temperature sensor connection to MBED NXP LPC1768:**

****

**LCD display connection to MBED:**

****

**Code :**

#include "mbed.h"

#include "TextLCD.h"

AnalogIn LM35(p15); //pin 15 of the mbed is used to read the analog output provided by the lm35 temperature sensor

TextLCD lcd(p5, p6, p7, p8, p9, p10); //pins for the functioning of LCD

int main() {

float temp=0.0; //variable used to store the calculated temperature

while(1) {

lcd.locate(0,1);

temp = ((LM35.read()\*3276)/10);

//printf("The Measured Temp is: %2.2f degree C\r\n", temp); //command to print on serial port monitor

lcd.printf("Temperature is: %2.2fC\r", result); //command to display the output on LCD Display

wait(1); //used to suspend execution of the calling thread

}

}

**Library Files:**

**“Mbed.h” -**

/\* mbed Microcontroller Library

\* Copyright (c) 2006-2013 ARM Limited

\* SPDX-License-Identifier: Apache-2.0

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* http://www.apache.org/licenses/LICENSE-2.0

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

#ifndef MBED\_H

#define MBED\_H

#include "platform/mbed\_version.h"

#if MBED\_CONF\_RTOS\_API\_PRESENT

#include "rtos/rtos.h"

#endif

#if MBED\_CONF\_NSAPI\_PRESENT

#include "netsocket/nsapi.h"

#include "netsocket/nsapi\_ppp.h"

#endif

#if MBED\_CONF\_EVENTS\_PRESENT

#include "events/mbed\_events.h"

#endif

#if MBED\_CONF\_FILESYSTEM\_PRESENT

#include "filesystem/mbed\_filesystem.h"

#endif

#include "platform/mbed\_toolchain.h"

#include "platform/platform.h"

#include "platform/mbed\_application.h"

// Useful C libraries

#include <math.h>

#include <time.h>

// mbed Debug libraries

#include "platform/mbed\_error.h"

#include "platform/mbed\_interface.h"

#include "platform/mbed\_assert.h"

#include "platform/mbed\_debug.h"

// mbed Peripheral components

#include "drivers/DigitalIn.h"

#include "drivers/DigitalOut.h"

#include "drivers/DigitalInOut.h"

#include "drivers/BusIn.h"

#include "drivers/BusOut.h"

#include "drivers/BusInOut.h"

#include "drivers/PortIn.h"

#include "drivers/PortInOut.h"

#include "drivers/PortOut.h"

#include "drivers/AnalogIn.h"

#include "drivers/AnalogOut.h"

#include "drivers/PwmOut.h"

#include "drivers/SPI.h"

#include "drivers/SPISlave.h"

#include "drivers/I2C.h"

#include "drivers/I2CSlave.h"

#include "drivers/CAN.h"

#include "drivers/RawCAN.h"

#include "drivers/UnbufferedSerial.h"

#include "drivers/BufferedSerial.h"

#include "drivers/FlashIAP.h"

#include "drivers/MbedCRC.h"

#include "drivers/QSPI.h"

#include "drivers/Watchdog.h"

// mbed Internal components

#include "drivers/ResetReason.h"

#include "drivers/HighResClock.h"

#include "drivers/Timer.h"

#include "drivers/Ticker.h"

#include "drivers/Timeout.h"

#include "drivers/LowPowerClock.h"

#include "drivers/LowPowerTimeout.h"

#include "drivers/LowPowerTicker.h"

#include "drivers/LowPowerTimer.h"

#include "drivers/RealTimeClock.h"

#include "platform/LocalFileSystem.h"

#include "drivers/InterruptIn.h"

#include "platform/mbed\_wait\_api.h"

#include "platform/mbed\_thread.h"

#include "hal/sleep\_api.h"

#include "platform/mbed\_atomic.h"

#include "platform/mbed\_power\_mgmt.h"

#include "platform/mbed\_rtc\_time.h"

#include "platform/mbed\_poll.h"

#include "platform/ATCmdParser.h"

#include "platform/FileSystemHandle.h"

#include "platform/FileHandle.h"

#include "platform/DirHandle.h"

#include "platform/CriticalSectionLock.h"

#include "platform/DeepSleepLock.h"

#include "platform/ScopedRomWriteLock.h"

#include "platform/ScopedRamExecutionLock.h"

#include "platform/mbed\_stats.h"

#include "platform/Stream.h"

// mbed Non-hardware components

#include "platform/Callback.h"

#include "platform/ScopedLock.h"

#ifndef MBED\_NO\_GLOBAL\_USING\_DIRECTIVE

using namespace mbed;

using namespace std;

#endif

#endif

**LINK -** <https://github.com/ARMmbed/mbed-os/#d147abc3e556c58e5e343d34b729bc2192e18bd3>

**“TextLCD.h” -**

/\* mbed TextLCD Library, for a 4-bit LCD based on HD44780

\* Copyright (c) 2007-2010, sford, http://mbed.org

\* Permission is hereby granted, free of charge, to any person obtaining a copy

\* of this software and associated documentation files (the "Software"), to deal

\* in the Software without restriction, including without limitation the rights

\* to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

\* copies of the Software, and to permit persons to whom the Software is

\* furnished to do so, subject to the following conditions:

\* The above copyright notice and this permission notice shall be included in

\* all copies or substantial portions of the Software.

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

\* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

\* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

\* AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

\* LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

\* OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

\* THE SOFTWARE.

\*/

#ifndef MBED\_TEXTLCD\_H

#define MBED\_TEXTLCD\_H

#include "mbed.h"

/\*\* A TextLCD interface for driving 4-bit HD44780-based LCDs

\* Currently supports 16x2, 20x2 and 20x4 panels

\* @code

\* #include "mbed.h"

\* #include "TextLCD.h"

\* TextLCD lcd(p10, p12, p15, p16, p29, p30); // rs, e, d4-d7

\* int main() {

\* lcd.printf("Hello World!\n");

\* }

\* @endcode

\*/

00043 class TextLCD: public Stream {

public:

/\*\* LCD panel format \*/

00047 enum LCDType {

00048 LCD16x2 /\*\*< 16x2 LCD panel (default) \*/

00049, LCD16x2B /\*\*< 16x2 LCD panel alternate addressing \*/

00050, LCD20x2 /\*\*< 20x2 LCD panel \*/

00051, LCD20x4 /\*\*< 20x4 LCD panel \*/

};

/\*\* Create a TextLCD interface

\* @param rs Instruction/data control line

\* @param e Enable line (clock)

\* @param d4-d7 Data lines for using as a 4-bit interface

\* @param type Sets the panel size/addressing mode (default = LCD16x2)

\*/

TextLCD(PinName rs, PinName e, PinName d4, PinName d5, PinName d6, PinName d7, LCDType type = LCD16x2);

#if DOXYGEN\_ONLY

/\*\* Write a character to the LCD

\* @param c The character to write to the display

\*/

int putc(int c);

/\*\* Write a formated string to the LCD

\* @param format A printf-style format string, followed by the

\* variables to use in formating the string.

\*/

int printf(const char \* format, ...);

#endif

/\*\* Locate to a screen column and row

\* @param column The horizontal position from the left, indexed from 0

\* @param row The vertical position from the top, indexed from 0

\*/

void locate(int column, int row);

/\*\* Clear the screen and locate to 0,0 \*/

void cls();

int rows();

int columns();

protected:

// Stream implementation functions

virtual int \_putc(int value);

virtual int \_getc();

int address(int column, int row);

void character(int column, int row, int c);

void writeByte(int value);

void writeCommand(int command);

void writeData(int data);

DigitalOut \_rs,

\_e;

BusOut \_d;

LCDType \_type;

int \_column;

int \_row;

};

#endif

**LINK -** <https://os.mbed.com/users/simon/code/TextLCD/docs/tip/TextLCD_8h_source.html>

**Output on Remote Device :**

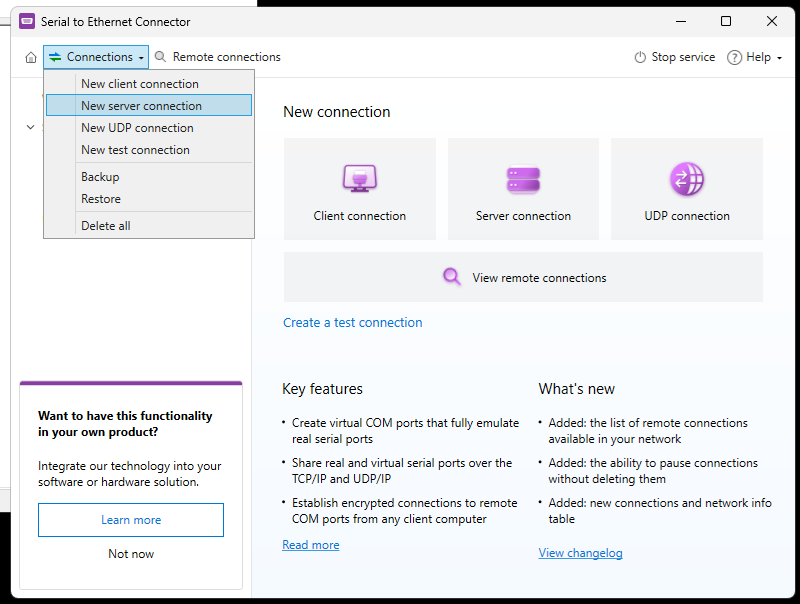
Application Used : Serial to Ethernet Connector

Link: <https://www.serial-over-ethernet.com/>

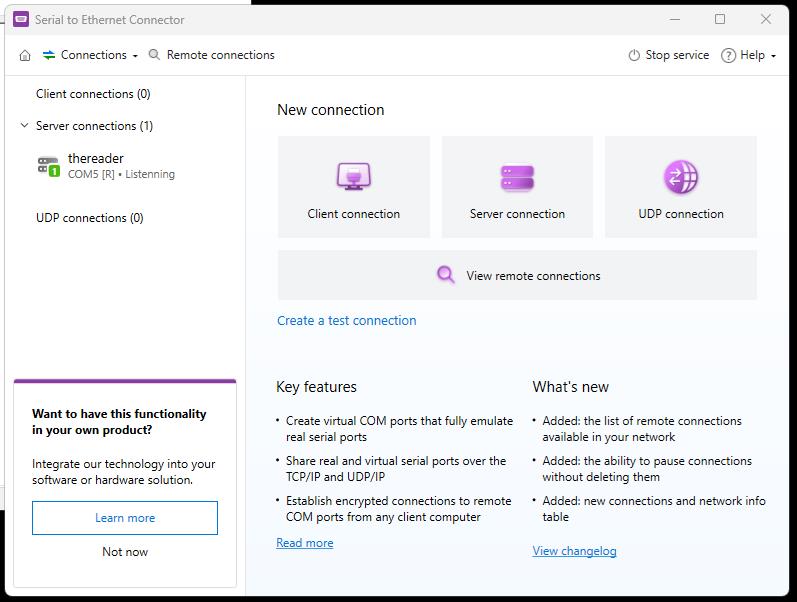
**Steps Performed :**

**Server Connection Creation:**

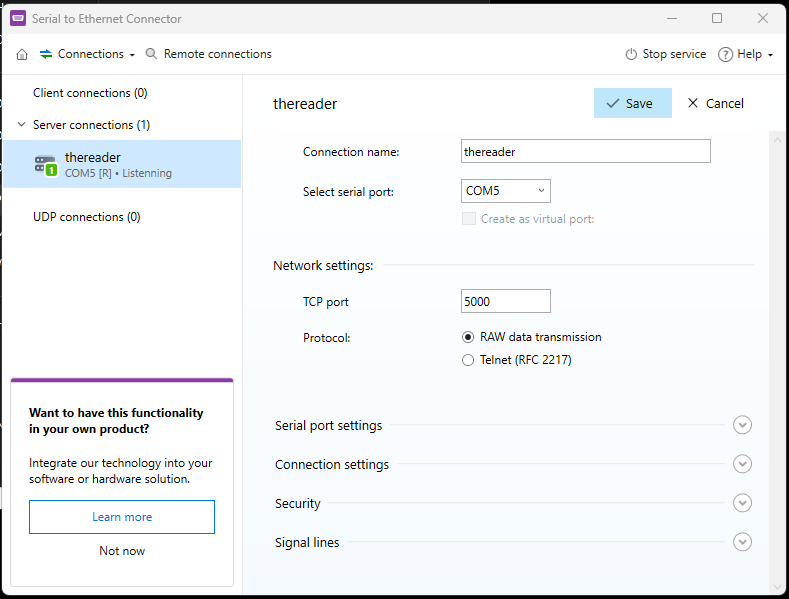
> The software is installed and launched.

****

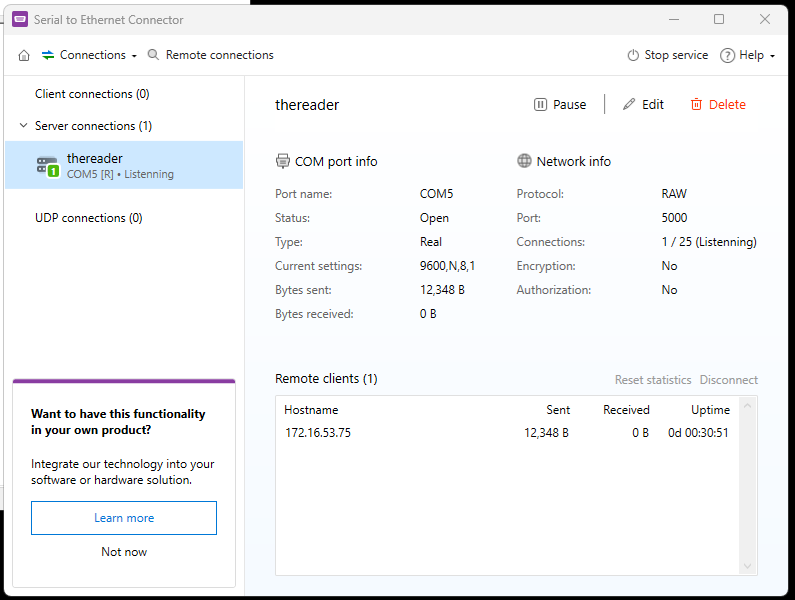
> Create a new server connection.

****

> Enter desired connection name and select serial port to which the mbed board is connected.

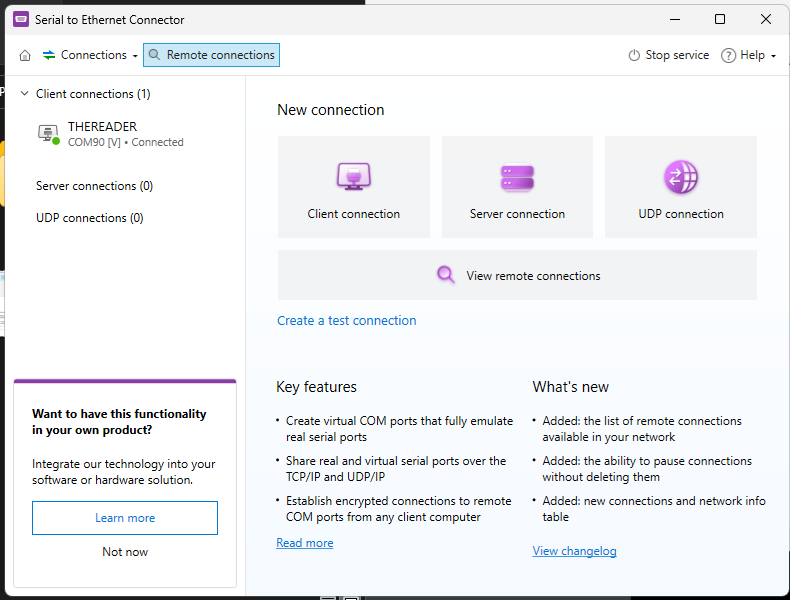
****

> Click save and now the server is created.

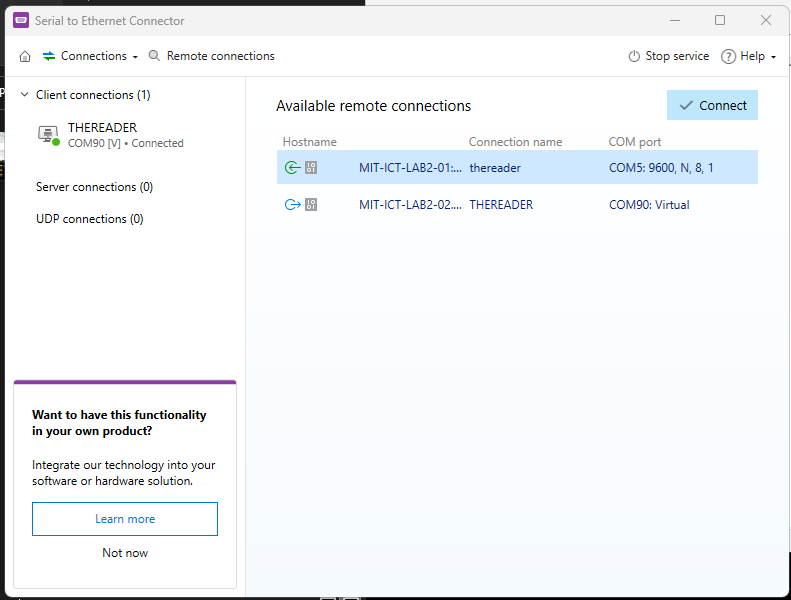
****

**Client Connection Creation:**

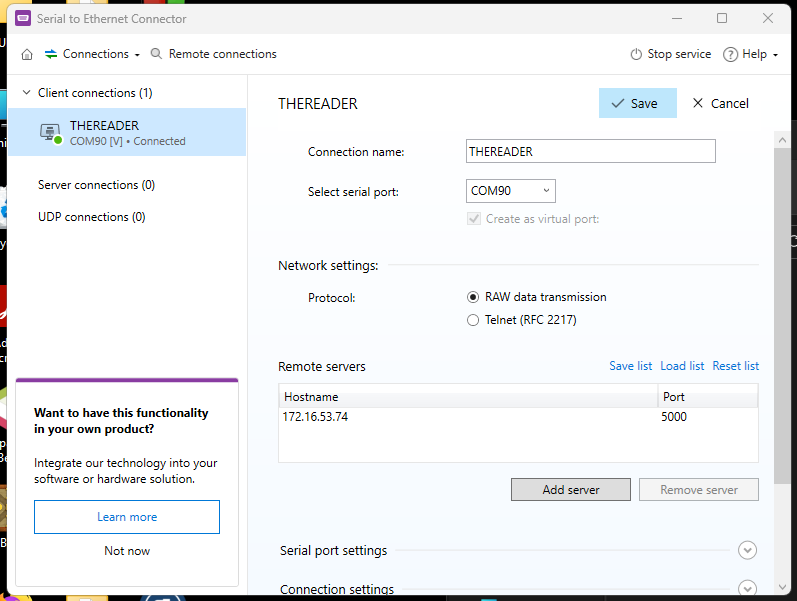
>On the remote PC, launch the application and Click on Remote Connection.

****

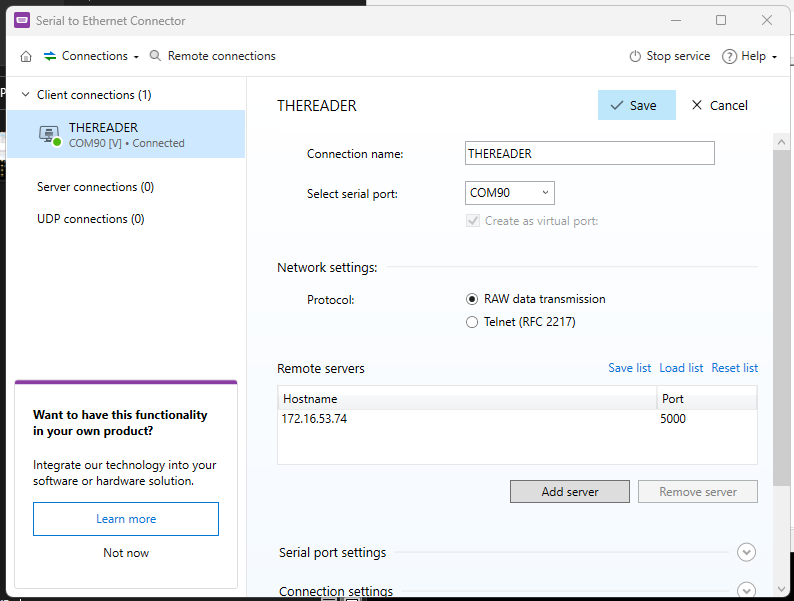
>The available remote connections should appear.

****

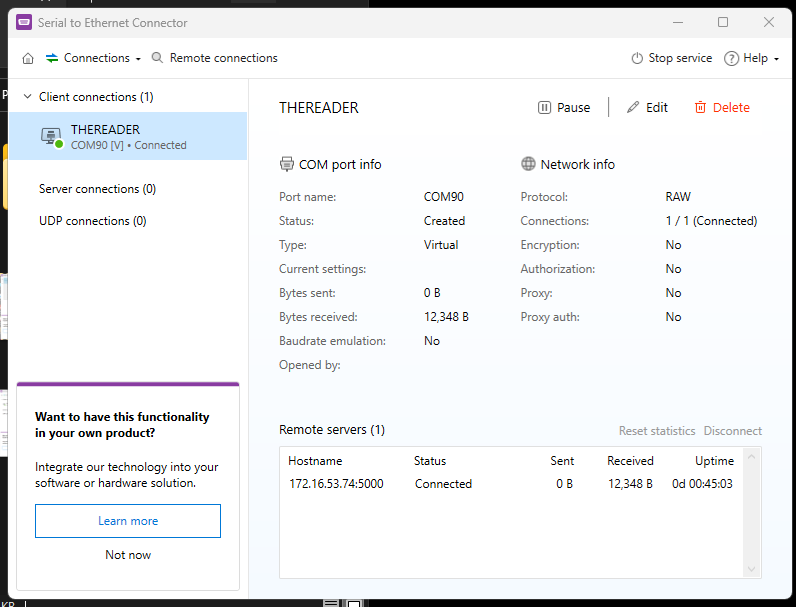
>Click on the server that you had created earlier.

****

>Make desired changes to the name and select the serial port on which the output should come. We have chosen COM90 in our demonstration.

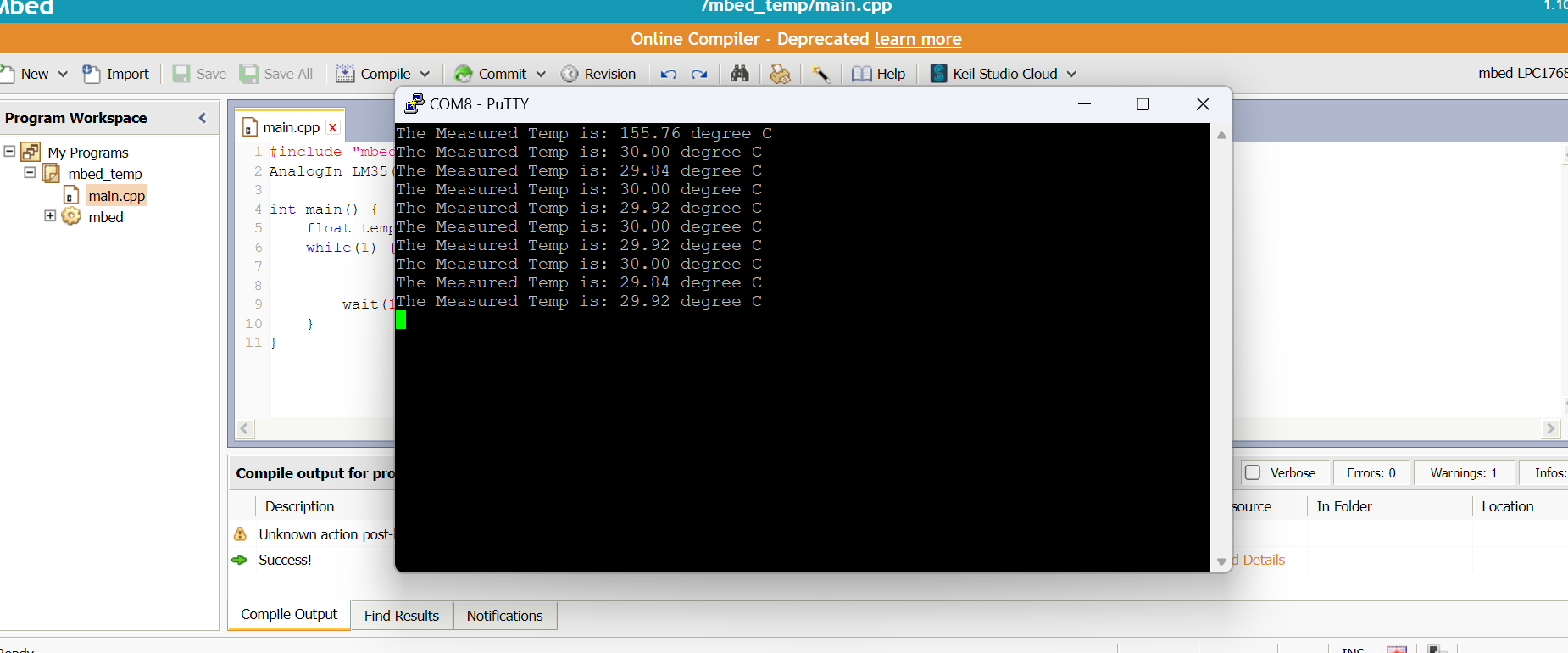
****

>Click on save and now the connection and port should get created.

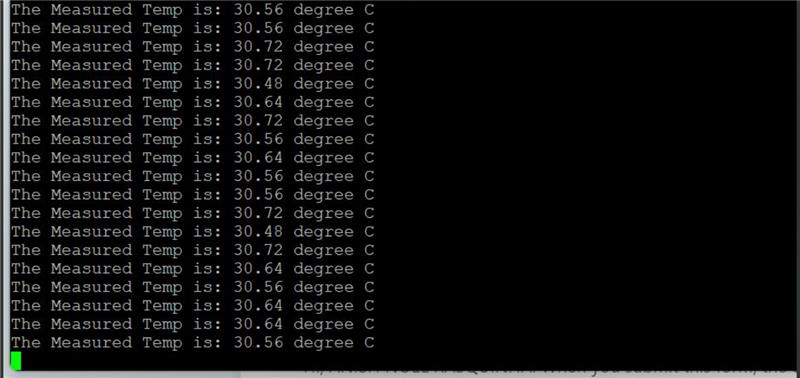
****

>Open putty on the remote device, choose serial connection type, choose the selected port (COM90 in our case) and click on open, which launches the terminal.

**Output :**

Output on terminal :

Output on remote device:



GDrive Link to Video demonstration :

[ES FISAC 2 video.mp4 - Google Drive](https://drive.google.com/file/d/10_P47k6jaTK-0gu_CizwPGdjyNv0c1VZ/view)

**Conclusion :**

By the end of this project we were able to induce ourselves with the proper working of the MBED board, its functionalities and its flaws. Deep knowledge about the working of the LM35 temperature sensor and LCD. Exploring new softwares to work hand in hand to bring it all together for the working of this project. This project let us deepen our knowledge about the vast varieties of header files which are required for the functioning of the code. ARM mbed boards being very compact and small and having tons of functionalities shows how powerful embedded systems are. Taught us about the importance of teamwork and having faith in our teammates to produce the best results possible.

**References :**

Github link to the mbed documentation : [GitHub - ARMmbed/mbed-os: Arm Mbed OS is a platform operating system designed for the internet of things](https://github.com/ARMmbed/mbed-os/#d147abc3e556c58e5e343d34b729bc2192e18bd3)

Serial port driver for windows installation instructions : [Windows serial configuration - Handbook | Mbed](https://os.mbed.com/handbook/Windows-serial-configuration)

PuTTY installation instructions : [Download PuTTY - a free SSH and telnet client for Windows](https://www.putty.org/)