University of Engineering and Technology, UET Taxila.

Electrical Engineering Department

(AT_L Project Report)

Submitted to: Engr. Tariq

Submitted by: Muhammad Saad (20-EE-103), Usama Hanif (20-EE-23), Hira Naseem (20-EE-07), Eesha Nasir (20-EE-71)

Section: C

Date: 15/12/2022

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Objective:

The Objective of this project is to build a pressure & temperature measuring device. The target is to build a system that can measure both pressure & temperature at the same time.

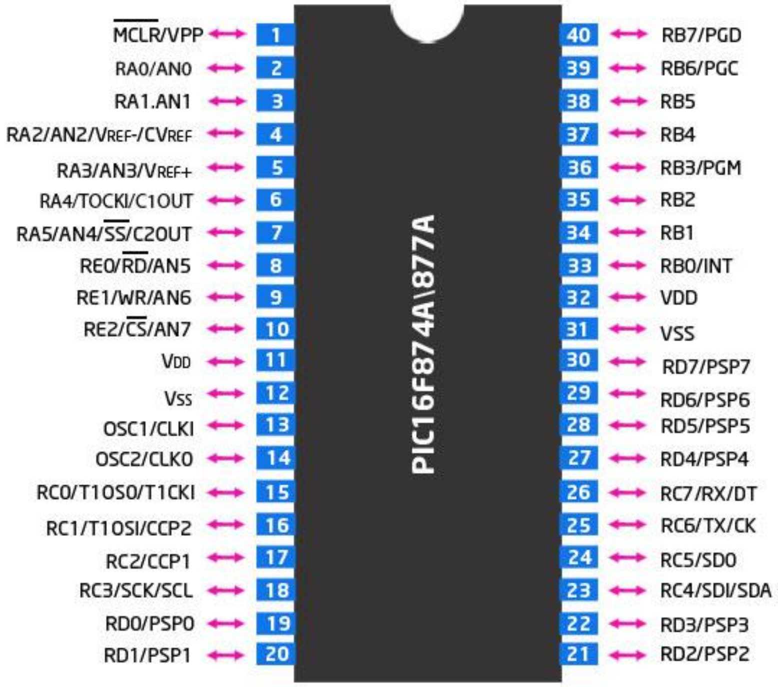
Equipment:

- Pic-16f877A (Micro-controller)
- 16x2 LCD
- 5V Voltage Regulator (7805)
- 9V DC Battery
- Vero Board
- Crystal Oscillator (8 MHz)
- Capacitors (22pF)
- Male Headers
- Jumper Wires

Theory:

(Pic-16f877A)

The PIC16F877A is a 40-pin (DIP) microcontroller which Microchip describes as powerful based on having a 200 nanosecond instruction speed. It's old and Microchip itself is not recommending it for new designs but its features and price make it still a popular microcontroller. PIC16F877A has 40 pins by 33 paths of I/O. The 40 pins make it easier to use the peripherals as the functions are spread out over the pins. This makes it easier to decide what external devices to attach without worrying too much if there are enough pins to do the job.

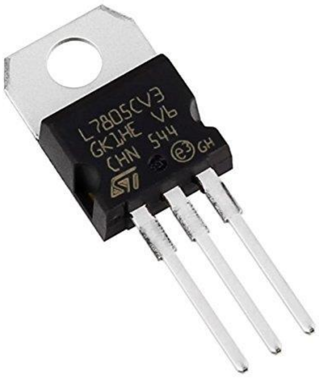


(16x2 LCD)

An LCD (Liquid Crystal Display) screen is an electronic display module and has a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. This LCD has two registers, namely, Command and Data.

(7805 Voltage Regulator)

Voltage regulators are very common in electronic circuits. They provide a constant output voltage for a varied input voltage. In our case the 7805 IC is an iconic regulator IC that finds its application in most of the projects. The name 7805 signifies two meaning, “78” means that it is a positive voltage regulator and “05” means that it provides 5V as output. So our 7805 will provide a +5V output voltage. The output current of this IC can go up to 1.5A. But, the IC suffers from heavy heat loss hence a Heat sink is recommended for projects that consume more current. For example if the input voltage is 12V and you are consuming 1A, then (12-5) * 1 = 7W. This 7 Watts will be dissipated as heat.



(9V DC Battery)

The battery is at 9V, and it pretty much keeps that voltage constant, until it dies. The chemical reactions inside the battery creates DC voltage. Electronic circuits really like DC voltage.

(Vero Board)

Veroboard is a brand of stripboard, a pre-formed circuit board material of copper strips on an insulating bonded paper board which was originated and developed in the early 1960s by the Electronics Department of Vero Precision Engineering Ltd (VPE). It was introduced as a general-purpose material for use in constructing electronic circuits - differing from purpose-designed printed circuit boards (PCBs) in that a variety of electronics circuits may be constructed using a standard wiring board.

(Crystal Oscillator)

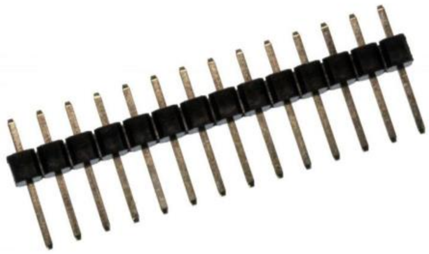
Crystal oscillators (resonators) are made from high-quality quartz crystal wafers. These wafers serve as the reference oscillator in microcontrollers. A crystal oscillator can vary in size, but thinner crystal cuts offer higher frequency operation. For example, 0.15 mm thick quartz crystal may operate at 15 MHz.

(Capacitors)

A capacitor is a device that is used to store charges in an electrical circuit. A capacitor works on the principle that the capacitance of a conductor increases appreciably when an earthed conductor is brought near it. Hence, a capacitor has two plates separated by a distance having equal and opposite charges.

(Male Headers)

Male headers are the simplest type and generally ship with any board that uses them. Use male headers if you want to build your project into a breadboard - solderless or permanent. They are also used in conjunction with female headers if you want to stack your boards together.



(Jumper Wires)

A jumper wire is an electric wire that connects remote electric circuits used for printed circuit boards. By attaching a jumper wire on the circuit, it can be short-circuited and short-cut (jump) to the electric circuit.

Construction:

The Project is soldered on a vero-board. Microcontroller is fitted in a holder and pins of holder are shorted with the male header for ease of using jumper wires. The crystal oscillator (8MHz) is used with microcontroller to provide frequency to it. Two capacitors of ratings 22pF are used with oscillator. LCD is attached with Port D of microcontroller. The Power is provided using 9V battery. This 9V battery is then attached to voltage regulator (7805). Then the whole module is power from the output of voltage regulator. BMP 280 is connected to microcontroller using I2C serial communication pins (SCL, SDA).

Code:

```
#include <stdint.h>
void BMP280_Write(uint8_t reg_addr, uint8_t _data) { I2C_Start(BMP280_STREAM);
  I2C_Write(BMP280_STREAM, BMP280_I2C_ADDRESS);
  I2C_Write(BMP280_STREAM, reg_addr);
  I2C_Write(BMP280_STREAM, _data);
  I2C_Stop(BMP280_STREAM);}
// BMP280 sensor configuration function
void BMP280_Configure(BMP280_mode mode, BMP280_sampling T_sampling,
BMP280_sampling P_sampling, BMP280_filter filter, standby_time standby){ uint8_t  _ctrl_meas, _config;
 _config = ((standby << 5) | (filter << 2)) & 0xFC;
_ctrl_meas = (T_sampling << 5) | (P_sampling << 2) | mode;
BMP280_Write(BMP280_REG_CONFIG,  _config);
BMP280_Write(BMP280_REG_CONTROL, _ctrl_meas);}
// initializes the BMP280 sensor, returns 1 if OK and 0 if error
int1 BMP280_begin(BMP280_mode mode,
BMP280_sampling T_sampling = SAMPLING_X1,
BMP280_sampling P_sampling = SAMPLING_X1,
BMP280_filter filter = FILTER_OFF,
standby_time  standby      = STANDBY_0_5)
{if(BMP280_Read8(BMP280_REG_CHIPID) != BMP280_CHIP_ID)
 return 0;
  // reset the BMP280 with soft reset
 BMP280_Write(BMP280_REG_SOFTRESET, 0xB6);
 delay_ms(100);
 // if NVM data are being copied to image registers, wait 100 ms
 while ( (BMP280_Read8(BMP280_REG_STATUS) & 0x01) == 0x01 )
  delay_ms(100);
  BMP280_calib.dig_T1 = BMP280_Read16(BMP280_REG_DIG_T1);
  BMP280_calib.dig_T2 = BMP280_Read16(BMP280_REG_DIG_T2);
  BMP280_calib.dig_T3 = BMP280_Read16(BMP280_REG_DIG_T3);
  BMP280_calib.dig_P1 = BMP280_Read16(BMP280_REG_DIG_P1);
  BMP280_calib.dig_P2 = BMP280_Read16(BMP280_REG_DIG_P2);
  BMP280_calib.dig_P3 = BMP280_Read16(BMP280_REG_DIG_P3);
  BMP280_calib.dig_P4 = BMP280_Read16(BMP280_REG_DIG_P4);
  BMP280_calib.dig_P5 = BMP280_Read16(BMP280_REG_DIG_P5);
  BMP280_calib.dig_P6 = BMP280_Read16(BMP280_REG_DIG_P6);
  BMP280_calib.dig_P7 = BMP280_Read16(BMP280_REG_DIG_P7);
  BMP280_calib.dig_P8 = BMP280_Read16(BMP280_REG_DIG_P8);
  BMP280_calib.dig_P9 = BMP280_Read16(BMP280_REG_DIG_P9);
  BMP280_Configure(mode, T_sampling, P_sampling, filter, standby);
  return 1;}
 // Temperature value is saved to *temp, returns 1 if OK and 0 if error.
 int1 BMP280_readTemperature(int32_t *temp){  int32_t var1, var2;
 BMP280_Update();
  // calculate temperature
  var1 = ((((adc_T / 8) - ((int32_t)BMP280_calib.dig_T1 * 2))) *
  ((int32_t)BMP280_calib.dig_T2)) / 2048;
  var2 = (((((adc_T / 16) - ((int32_t)BMP280_calib.dig_T1)) *
  ((adc_T / 16) - ((int32_t)BMP280_calib.dig_T1))) / 4096) *
  ((int32_t)BMP280_calib.dig_T3)) / 16384;
  t_fine = var1 + var2;
  *temp = (t_fine * 5 + 128) / 256;
  return 1;}
// Reads pressure from BMP280 sensor.
// Pressure is stored in Pa (output value of "96386" equals 96386 Pa = 963.86 hPa).
// Pressure value is saved to *pres, returns 1 if OK and 0 if error.
```

```
int1 BMP280_readPressure(uint32_t *pres){  int32_t var1, var2;
uint32_t p;
// calculate pressure
var1 = (((int32_t)t_fine) / 2) - (int32_t)64000;
var2 = (((var1/4) * (var1/4)) / 2048 ) * ((int32_t)BMP280_calib.dig_P6);
var2 = var2 + ((var1 * ((int32_t)BMP280_calib.dig_P5)) * 2);
var2 = (var2/4) + (((int32_t)BMP280_calib.dig_P4) * 65536);
var1 = ((((int32_t)BMP280_calib.dig_P3 * (((var1/4) * (var1/4)) / 8192 )) / 8) +
((((int32_t)BMP280_calib.dig_P2) * var1)/2)) / 262144;
var1 =((((32768 + var1)) * ((int32_t)BMP280_calib.dig_P1)) / 32768);
if (var1 == 0)
return 0; // avoid exception caused by division by zero
p = (((uint32_t)(((int32_t)1048576) - adc_P) - (var2 / 4096))) * 3125;
if (p < 0x80000000)
p = (p * 2) / ((uint32_t)var1);
else
p = (p / (uint32_t)var1) * 2;
var1 = (((int32_t)BMP280_calib.dig_P9) * ((int32_t)(((p/8) * (p/8)) / 8192))) / 4096;
var2 = (((int32_t)(p/4)) * ((int32_t)BMP280_calib.dig_P8)) / 8192;
p = (uint32_t)((int32_t)p + ((var1 + var2 + (int32_t)BMP280_calib.dig_P7) / 16));
*pres = p;
return 1;}
```

Working:

The BMP 280 sensor sense the temperature and pressure and transform into electrical form (differences of voltages) and send it to Pic-16f877a microcontroller using I2C communication. The microcontroller receive the data in form of electrical signals process it and obtain equivalent numerical values. These values are then displayed on the 16x2 LCD. Module take 9V input from the battery and the convert it into 5V DC through 7805 voltage regulator. Then all the parts including, LCD, sensor & microcontroller is power up using the output of voltage regulator.

Conclusion:

We made a module which can measure atmospheric pressure and temperature at the same time. It shows us the variation in the both parameters with great accuracy. One can monitor temperature or pressure of environment using this module.