|  |  |
| --- | --- |
| Experiment No: 4 | Sensors and Analog to Digital Converter (ADC) |

**Objectives:**

To learn about sensors and how to connect them to PIC 16f877a and do some application

**Analog to Digital Converter (ADC).**

In nature, there are various electrical signals like analog, which means the time-varying continuous signal. In such signals, the first quantity of the signal is a voltage, whereas the next quantity of the signal can be anything like pressure, force, temperature and light accelerations. Most of [the microcontrollers](https://www.watelectronics.com/8051-microcontroller-architecture/) available in the market are digital. They can only recognize high or low-level input pins. For example, if the input is greater than 2.5 volts, then it will be treated as high, and if it is less than 2.5 volts, then it will be treated as low. So, we cannot directly measure voltage from microcontrollers.

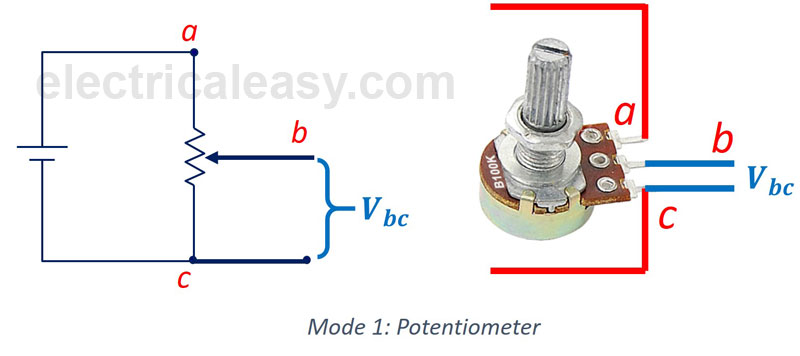
To overcome this problem, microcontrollers use an [A/D (analog to digital) conversion unit](https://www.elprocus.com/analog-to-digital-adc-converter/) that helps change a voltage to a number. This permits to interface all kinds of analog devices with microcontroller units. Some of the analog devices are light, temperature and touch devices, microphones and accelerometers.

The most important requirement of analog-to-digital conversion is the resolution. This states how exactly the A/D conversion measures the analog i/p signals. The common A/D conversions available in the market are 8-bit, 10-bit &12-bit. For example, if the reference voltage of A/D conversion is 0-5V, then a 8-bit A/D converter breaks this voltage into a number of parts (256 parts). So it can calculate the voltage exactly up to 5/256v= 19 mV approx. On the other hand, a 10-bit A/D converter breaks the voltage into 1024 parts. Hence, this converter can calculate the voltage accurately up to 5/1024= 4.8mV approx. Therefore, you can notice that the 8-bit A/D converter cannot tell the difference between 1mV & 18mV. The A/D converter in PIC microcontroller is 10-bit.

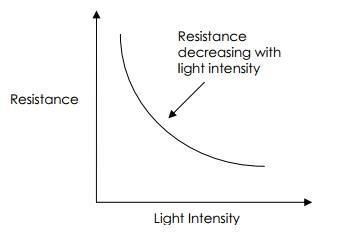
**Sensors:**

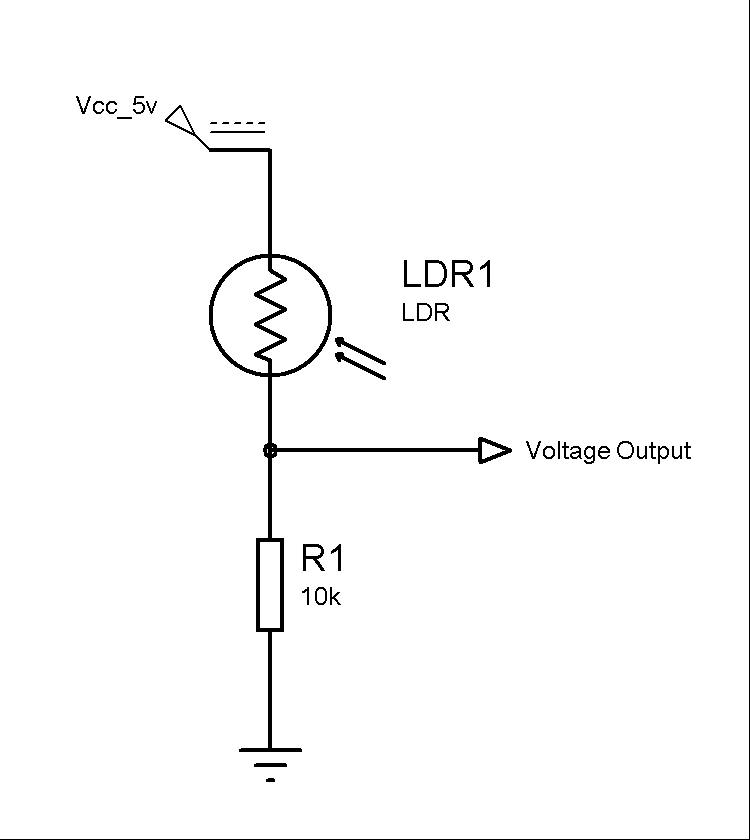
1. **Potentiometer** is a three-[terminal](https://en.wikipedia.org/wiki/Terminal_(electronics)) [resistor](https://en.wikipedia.org/wiki/Resistor) with a sliding or rotating contact that forms an adjustable [voltage divider](https://en.wikipedia.org/wiki/Voltage_divider). If only two terminals are used, one end and the wiper, it acts as a [variable resistor](https://en.wikipedia.org/wiki/Potentiometer#Rheostat) or [rheostat](https://en.wikipedia.org/wiki/Potentiometer#Rheostat).





1. **LDR** is a component that has a (variable) resistance that changes with the light intensity that falls upon it. This allows them to be used in light-sensing circuits





1. **LM35** series are precision integrated-circuit temperature devices with an output voltage linearly proportional to the Centigrade temperature and have the following features :

• Calibrated Directly in Celsius (Centigrade)

• Linear + 10-mV/°C Scale Factor

• 0.5°C Ensured Accuracy (at 25°C)

• Rated for Full −55°C to 150°C Range

• Suitable for Remote Applications

• Low-Cost Due to Wafer-Level Trimming

• Operates From 4 V to 30 V

• Less Than 60-μA Current Drain

• Low Self-Heating, 0.08°C in Still Air

• Non-Linearity Only ±¼°C Typical

• Low-Impedance Output, 0.1 Ω for 1-mA Load

****

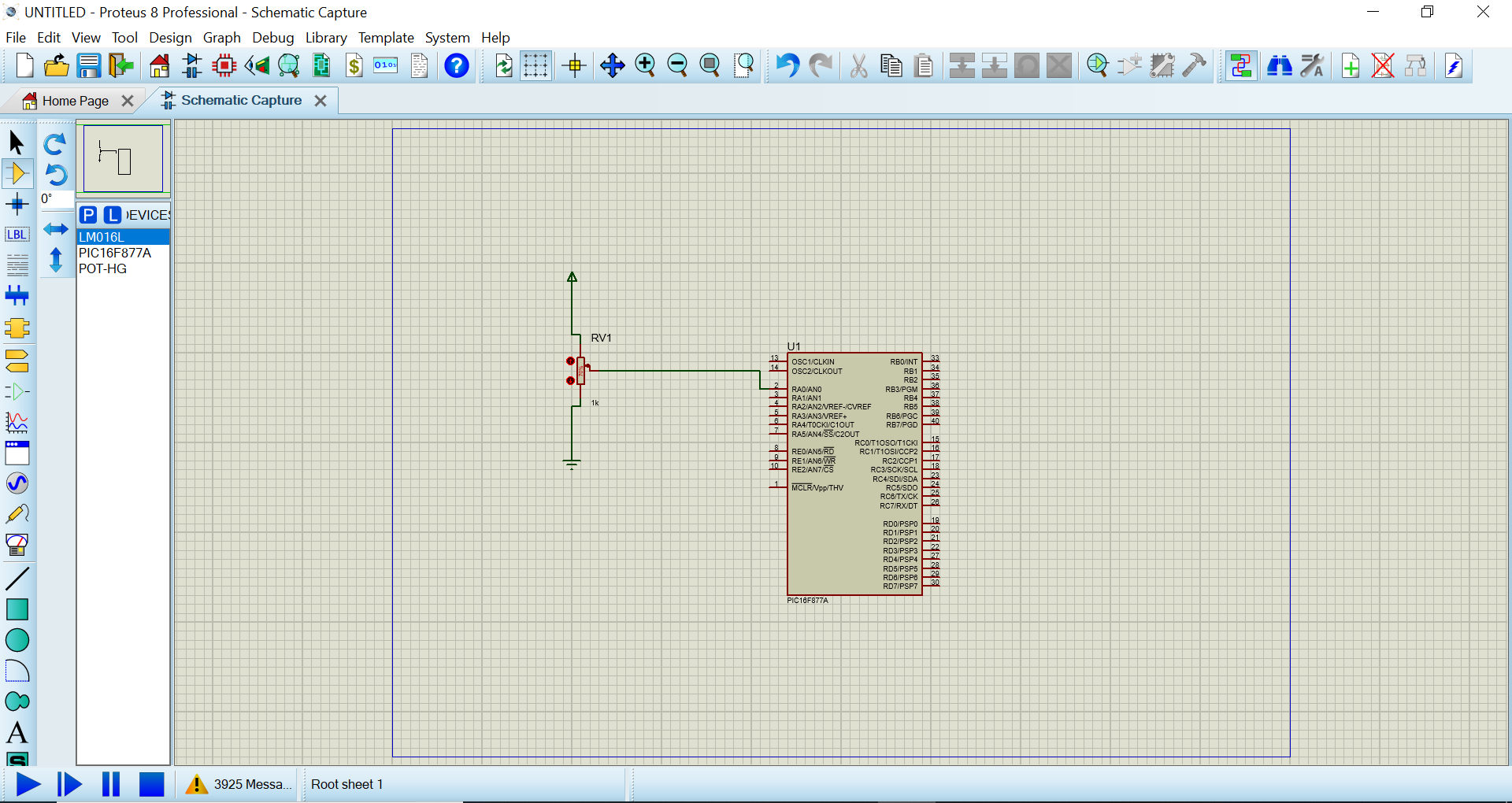
**Example 1**: Analog to Digital Converter (ADC).

Show the analog to digital result on the port b

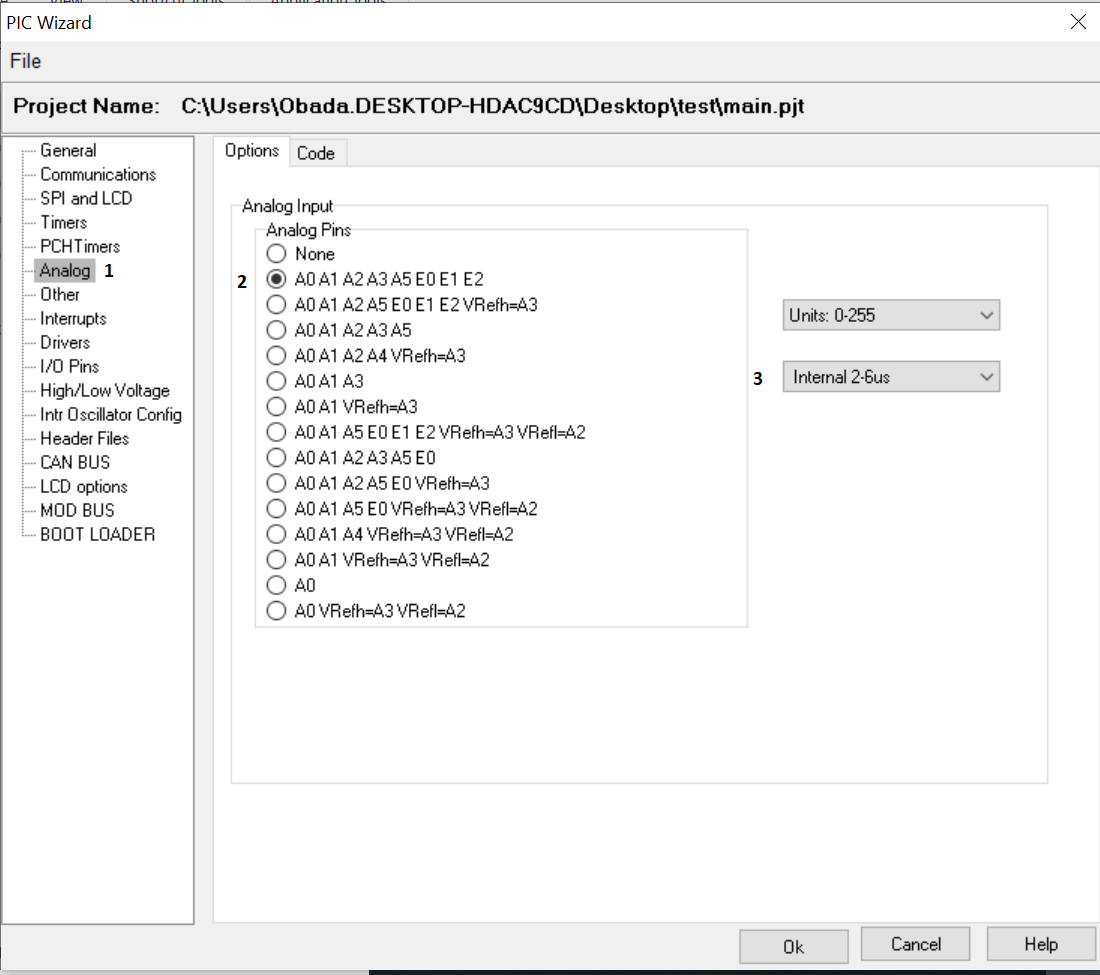
**Components:**

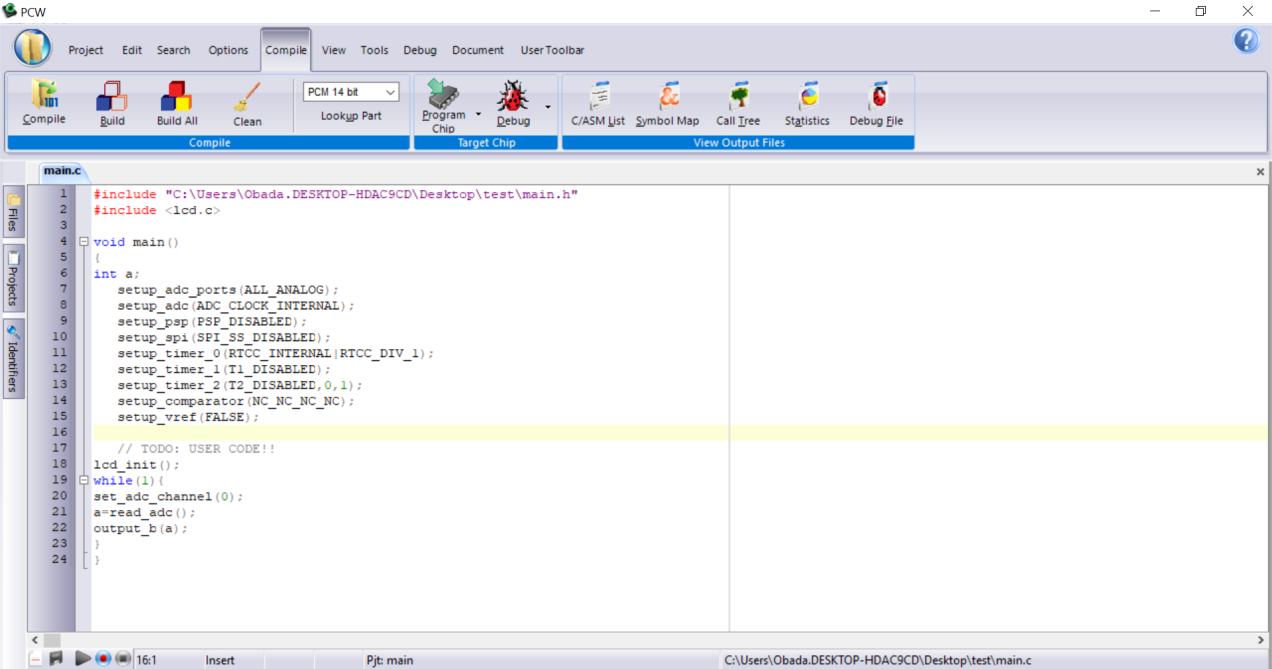
1. PIC16f877a form (Component mode)
2. Pot-hg (Component mode)

**Proteus:**

****

**PIC C**: from pic wizard choose analog as 1 in the picture below then chose 2 this will activate all the analog pins then from 3 choose internal 2-6us, the first step in programming it to activate the sensor channel using the command set\_adc\_channel(0); where 0 is referring to the first channel which represents a0, then use the function read\_adc to save the result from the analog to digital conversion in variable a; the value which is stored in a is given on the port b.





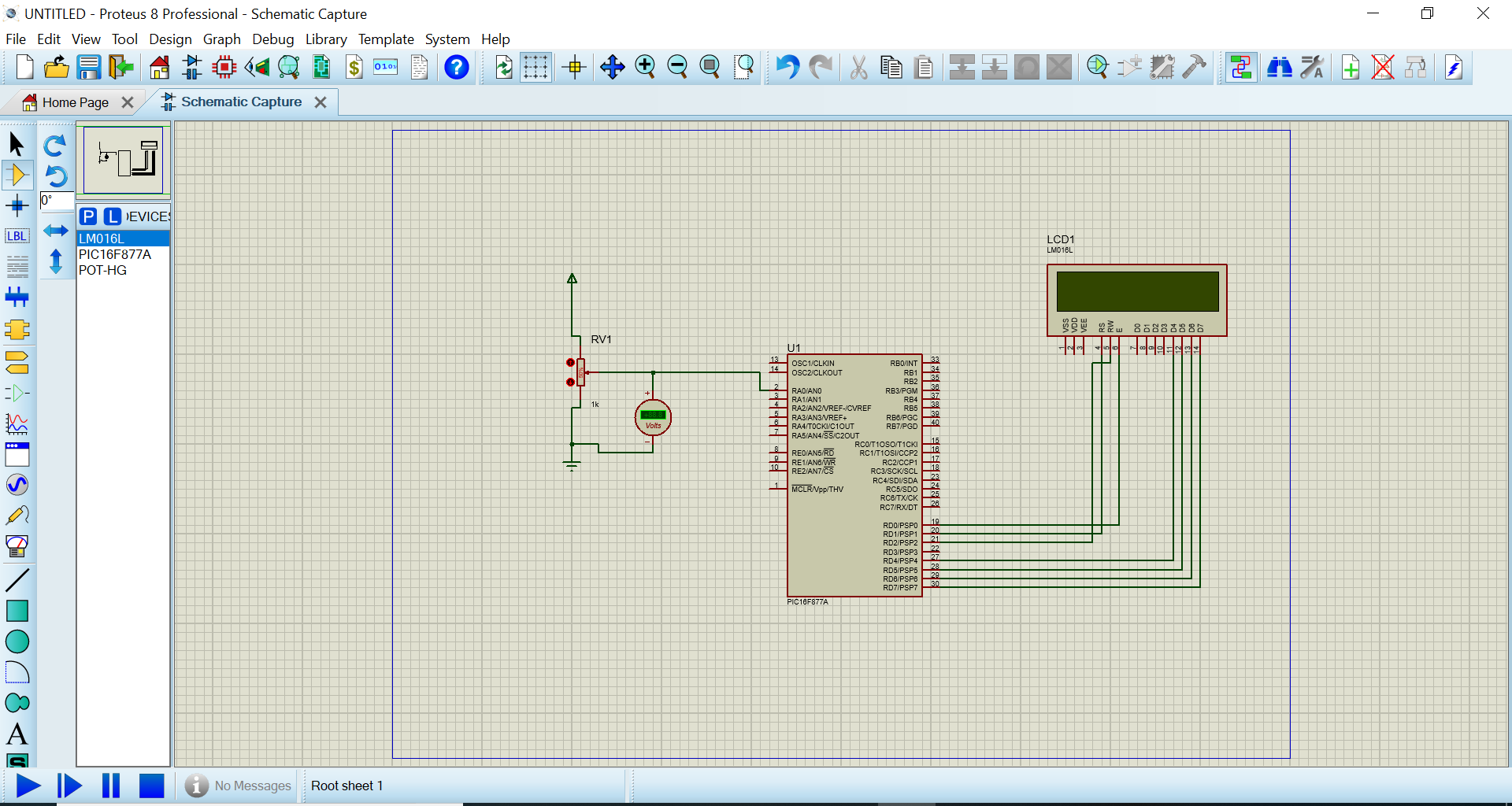
**Example 2**: Measuring the voltage of the potentiometer

The goal of this example is to measure the output voltage from the potentiometer.

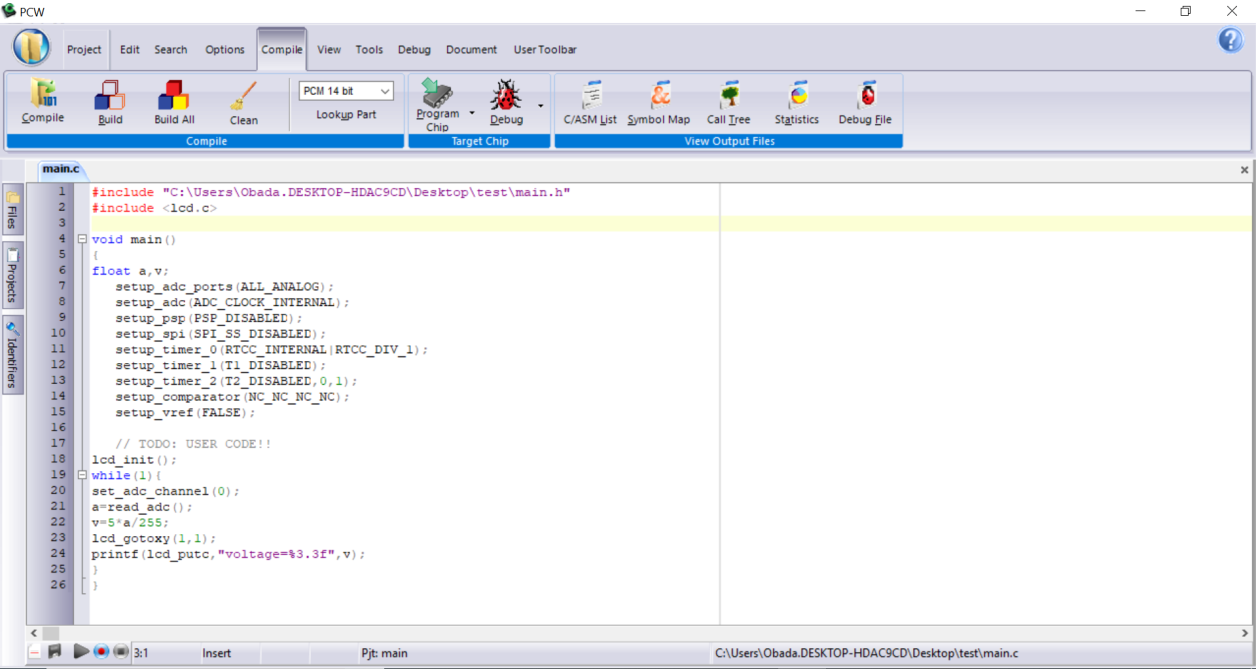
**Components:**

1. PIC16f877a form (Component mode)
2. LM016L (Component mode)
3. Pot-hg (Component mode)
4. Dc voltmeter (Instruments)

**Proteus:**



**PIC C**: as in the previous example channel 0 is activated then using variable a to store the result from conversion the convert it into voltage using the relation v=5\*a/255 , finally the result is shown on the LCD using the command printf



**Example 3:** Measuring the voltage of the potentiometer (more than 5 v)

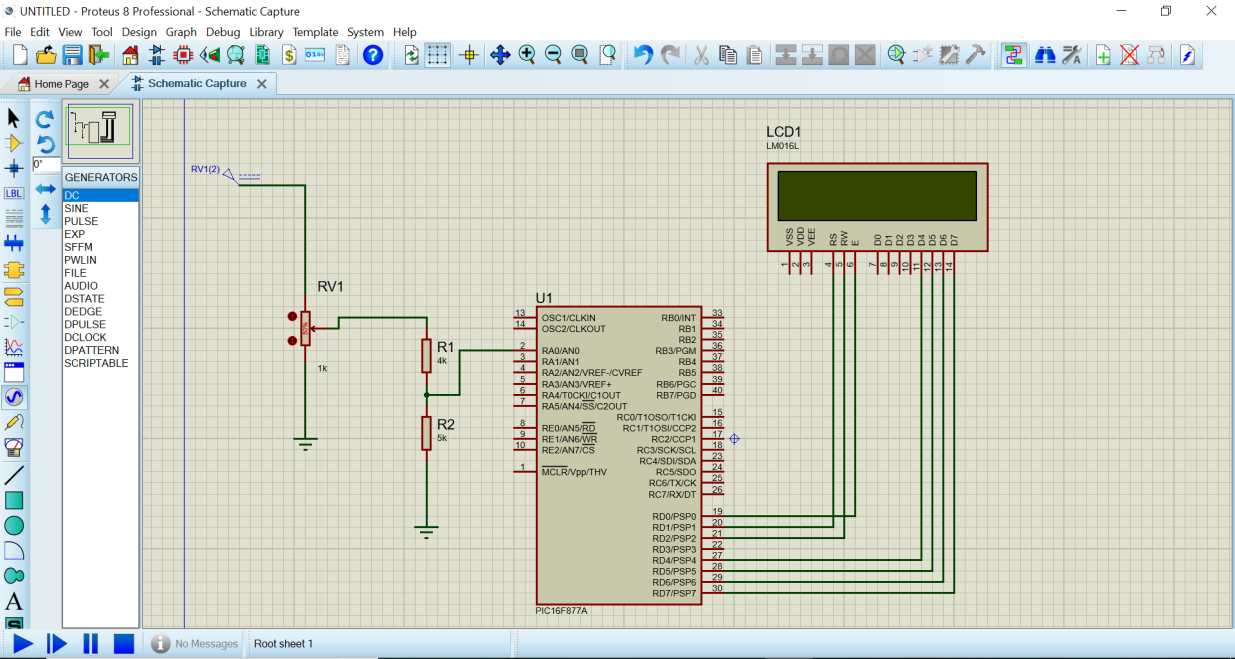
The goal of this example is to measure the output voltage from the potentiometer when the voltage supply is more than 5 v.

**Components:**

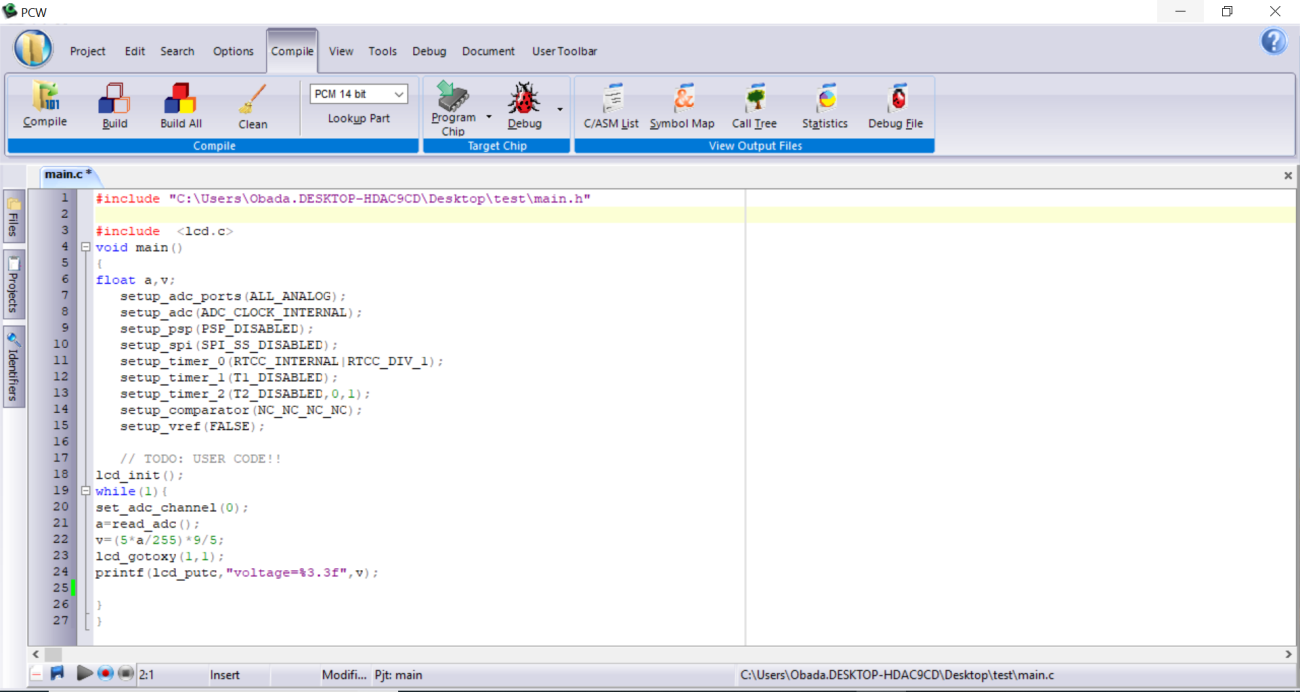
1. PIC16f877a form (Component mode)
2. LM016L (Component mode)
3. Pot-hg (Component mode)
4. Dc voltmeter (Instruments)
5. Res (Component mode)
6. DC (Generator mode)

**Proteus:**

By click on the DC change the voltage into 9 , change the values for resistance into 4k ohm and 5k ohm as below



PIC C: as per the previous example the voltage is multiplied by factor 9/5 which represents the maximization of voltage which it is reduced by using hardware (voltage divider) the goal of this is prevent to the connection to the pin analog voltage of more than 5 volts.



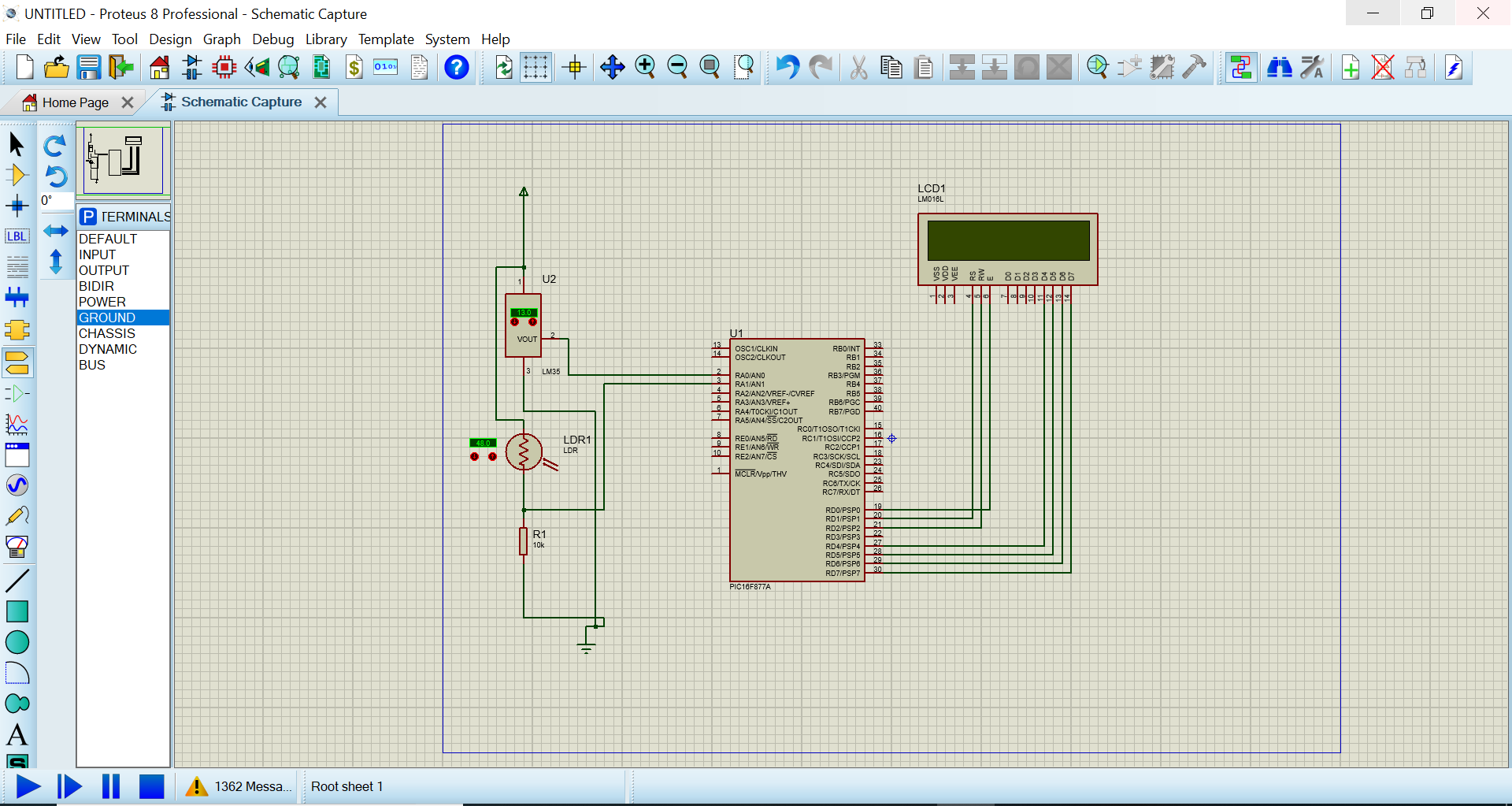
**Example 4:** Measuring the temperature and lux intensity

The goal of this example is to measure the temperature and lux intensity using lm35 and LDR sensors respectively.

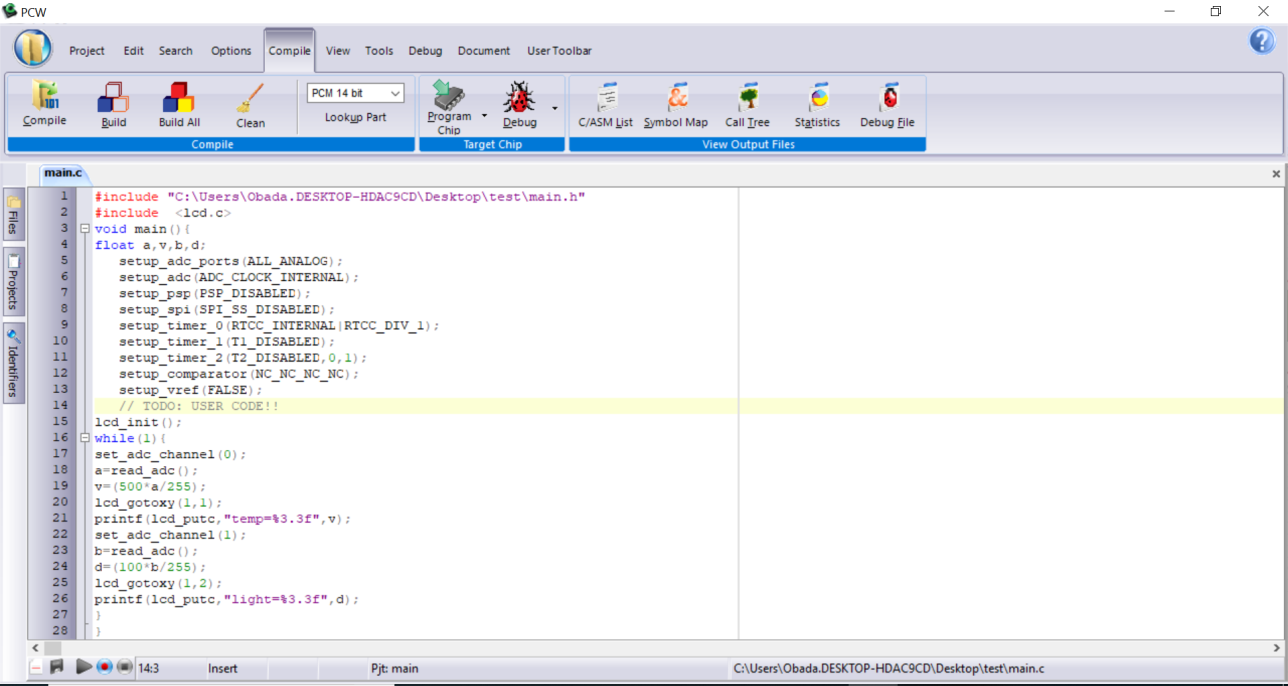
**Components:**

1. PIC16f877a form (Component mode)
2. LM016L (Component mode)
3. LM35 (Component mode)
4. LDR (Instruments)
5. Res (Component mode)

**Proteus:**



**PIC C:**

****