#include "L4\_123.h"

#define F\_CPU 16000000

#include <asf.h>

#include <stdio.h>

#include <avr/delay.h>

#include <avr/io.h>

#include <avr/interrupt.h>

#define \_CRT\_SECURE\_NO\_WARNINGS

//Parity 0

#define OFF\_PARITY 0

#define ODD\_PARITY 1

#define EVEN\_PARITY 2

#define BIT\_FORMAT\_8bit 1

//StopBit

#define STOP\_BIT\_1bit 1

#define STOP\_BIT\_2bit 2

char\* ReadString( unsigned port);

int WriteString(unsigned port, char \*szOutput);

//Task 4 Functions = Task 1 Functions

//------------------------------------------------

int ADC\_Read(void)

{

int result = 0;

// reset the converter

ADCSRA |= (1<<ADSC); //

while(!(ADCSRA & (1 << ADIF))); // ADIF turns on after reset

ADCSRA |= (1<<ADSC); // start conversion

while(!(ADCSRA & (1 << ADIF))); // ADIF turns on after conversion

result = ADCL; // read 1st lower part

result += (ADCH << 8); // read upper part

//ADCSRA &= ~(1<<ADEN); // AD-converter off

return result;

}

void ADC\_Init(char port)

{

if (port < 15)

{

ADMUX = (1 << REFS1) | (1 << REFS0) | port;

// set frequency 1MHz / 8 = 125kHz

ADCSRA = (1<<ADPS1) | (1<<ADPS0); //prescale

ADCSRA |= 1 << ADEN; // ADC Enable

}

}

//------------------------------------------------

int main (void)

{

int result; // variale to read integer value from ADC

int i=0; // variable to initialize ADC ports in a loop

double voltage; // double type value to convert ADC value

char string[30]="ADC starts.. "; // print the string in the beginning

SetLineParameters( 0, 9600, EVEN\_PARITY, BIT\_FORMAT\_8bit, STOP\_BIT\_2bit ); // init board

WriteString((unsigned) 0, string); // write the started string in the terminal

// Task 4

while (1) // infinite loop

{

for (i=0;i<8;i++)

{

ADC\_Init(i); // initialize the new port

result=ADC\_Read(); // read a new port

voltage=(double)result; // convert the integer value to double

voltage=(voltage\*2.5)/1023; // convert to decimal

sprintf(string,"\r\nResult ADC%i = %.5f Volts",i, voltage); // set to the string

WriteString((unsigned) 0, string); // show the string on a screen

}

\_delay\_ms(10000); // wait 10 seconds

WriteString((unsigned) 0, "\033[2J"); // clear the terminal window for new values

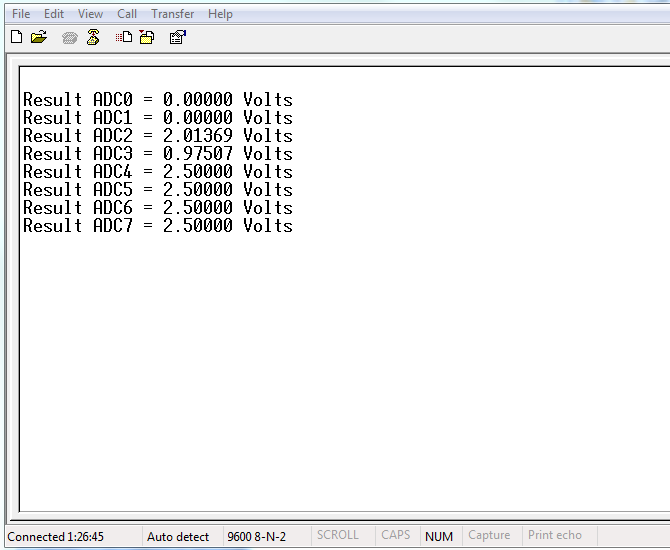
}

}

**RESULT**

**Picture 1**

ADC0, ADC1 are Grounded; ADC2, ADC3 have voltages from the power source and the rest ADCs pins are not grounded, that’s why they have value equals to Vref.



**Picture 2**

We can change inputs values during 10 seconds no changes will be shown, but after 10 seconds new values will be shown in the terminal. ADC0, ADC3 are Grounded; ADC1, ADC2 have voltages from the power source and the rest ADCs pins are not grounded, that’s why they have value equals to Vref.

