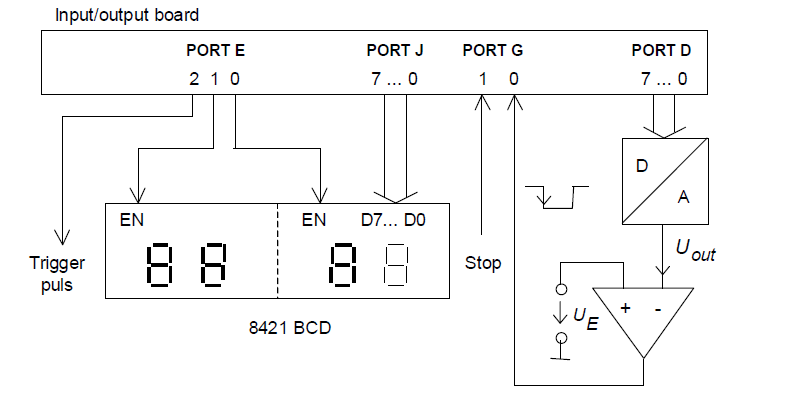
|  |  |  |
| --- | --- | --- |
| **Digital Voltmeter** | | |
| **Lab No:** *03* |  | **Chairperson:**  *Isuru Udana Abeysinghe* |
| **Date:** *02.12.2013* | **Participants:**  *Gervais Mckoy*  *Sukrat Khanna* |
| **Professor:**  *Dr.Lutz Leutelt* |

Experiment:

The task was to implement a c code which converts digital input signals to analog. This software was transferred to the microcontroller (**LM3S9B92**) and the result was displayed using **8421 BCD** display.

\*The circuit diagram is shown below



**External D/A converter**

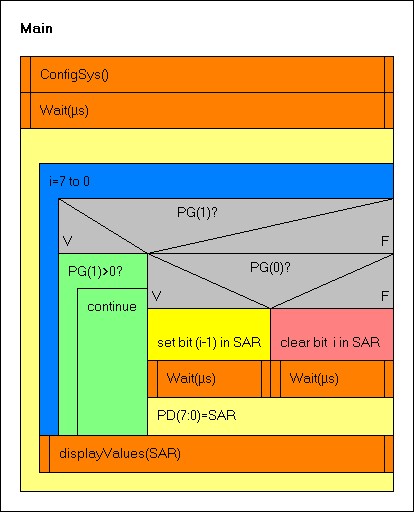
**Weighting conversion method (successive approximation)**

The task was to display the current voltage value using the 8421 BCD display. The hint in lab exercise was used to implement the c code.

In the program the digital numbers were generated via a “for loop”. For every instance in the loop the resulting voltage UOUT was compared with the analog input voltage UE. If UE > UOUT the voltage level at PG (0) was set to high, else the voltage level was set to low by the comparator. Conversion was stopped when UOUT  = UE. Afterwards this value was sent to the BCD display via port J and E.

\*\*When the digital input is changed, there is a delay of about 30μs until the comparator output is stable. This was taken into consideration for the development of the c code.

**Nassi-Shneiderman diagram.**



#define PART\_LM3S9B92

#include "lm3s9b92.h"

#include "stdio.h"

void wait(unsigned short usec)

{

TIMER0\_TAILR\_R=((16000000 /8)\* (usec/1000)); // ILR= 16M/122\*0.5-1

TIMER0\_CTL\_R |= 0x0001; // enable Timer 0

while((TIMER0\_RIS\_R&(1<<0))==0);

TIMER0\_ICR\_R=0x01;

TIMER0\_CTL\_R &= ~0x0001; // disable Timer \*/

}

void configSys(void)

{

int i=0;

SYSCTL\_RCGC2\_R |= (1<<3)| (1<<6)| (1<<8)| (1<<4); // PD,PG,PJ,PE

SYSCTL\_RCC\_R =((SYSCTL\_RCC\_R |0x00000540)&~ 0x000002B1);

//PORTD

GPIO\_PORTD\_DEN\_R = 0xFF;

GPIO\_PORTD\_DIR\_R = 0xFF;

//PORTE

GPIO\_PORTE\_DEN\_R |=0x07;

GPIO\_PORTE\_DIR\_R |=0x07;

//PORTG

GPIO\_PORTG\_DEN\_R = 0x03;

GPIO\_PORTG\_DIR\_R = 0x00;

//PORTJ

GPIO\_PORTJ\_DEN\_R = 0xFF;

GPIO\_PORTJ\_DIR\_R = 0xFF;

//Timer Configuration

SYSCTL\_RCGC1\_R |= (1<<16);

i++;

TIMER0\_CTL\_R &= ~0x0001; // disable Timer 0

TIMER0\_CFG\_R = 0x04; // 2 x 16-bit mode

TIMER0\_TAMR\_R = 0x22; // periodic mode + match enable

TIMER0\_TAPR\_R = 7; // pre-scaler PR= ceil((16M/2^16)\*0.03) -1}

void displayValue(int SAR)

{

int total, temp1, dig1,dig2, buyTime=0;

GPIO\_PORTE\_DATA\_R= 0x01;

buyTime++;

total=(int)(SAR \* 19.53125);

temp1=total % 100;

GPIO\_PORTJ\_DATA\_R= (int)(((temp1/10)<<4)|(temp1%10));

buyTime++;

total/= 100;

GPIO\_PORTE\_DATA\_R= 0x02;

buyTime++;

GPIO\_PORTJ\_DATA\_R= (int)(((total/10)<<4)|(total%10));

//GPIO\_PORTJ\_DATA\_R=(int)((dig2<<4)|dig1);

}

void main(void)

{

int i, SAR;

configSys();

while(1)

{

SAR= 0x80;

GPIO\_PORTE\_DATA\_R=0x04;

GPIO\_PORTE\_DATA\_R=0x00;

for(i=7; i>0;i--) //starting from the bit in the 7th position

{

if(GPIO\_PORTG\_DATA\_R & 0x02)

{

while(GPIO\_PORTG\_DATA\_R>1);

}

else

{

GPIO\_PORTD\_DATA\_R=SAR; //giving the value of SAR for

PORTD

wait(1); //allow PORTD to be updated

if(GPIO\_PORTG\_DATA\_R & 0x01) //comparing the inputs via

comparator

SAR|=(1<<i-1); //setting the i-1 bit to 1

else

{

SAR &=~(1<<i); //clearing the ith bit

SAR |=(1<<i-1);// setting the i-1 bit to 1

}

}

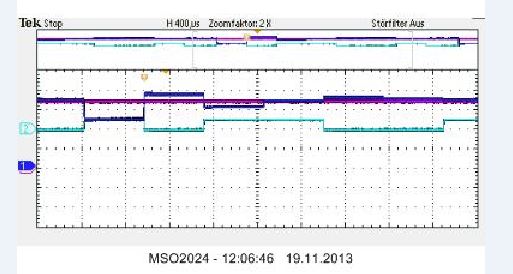
}

displayValue(SAR); //sending the value to be displayed

}

}

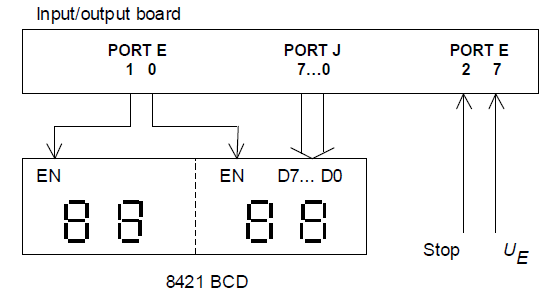
Corresponding Oscilloscope graph



**Internal A/D converter**

In this experiment the task was to convert the analog input into a digital value. Therefor Port E (7) was connected to an analog constant voltage of 5V. The output voltage was displayed using ports E and J. And the result was compared using Digital Multi Meter (DMM).

\*\*The circuit diagram is shown below.



#define PART\_LM3S9B92

#include "lm3s9b92.h"

#include "stdio.h"

void wait(unsigned short usec)

{

int i;

TIMER0\_TAILR\_R=((16000000 /8)\* (usec/1000)); // ILR= 16M/122\*0.5-1

TIMER0\_CTL\_R |= 0x0001; // enable Timer 0

while((TIMER0\_RIS\_R&(1<<0))==0);

TIMER0\_ICR\_R=0x01;

TIMER0\_CTL\_R &= ~0x0001; // disable Timer

}

void configSys(void)

{

int i;

SYSCTL\_RCGC2\_R |= (1<<3)| (1<<6)| (1<<8)| (1<<4); // PD,PG,PJ,PE

SYSCTL\_RCGC0\_R |= (1<<16); //ADC0

SYSCTL\_RCC\_R =((SYSCTL\_RCC\_R |0x00000540)&~ 0x000002B1);

//PORTE

GPIO\_PORTE\_DIR\_R |=0x07;

GPIO\_PORTE\_DIR\_R &=~0x04;

// configure AIN0 (=PE(7)) as analog inputs

GPIO\_PORTE\_DEN\_R |=0x07;

GPIO\_PORTE\_AFSEL\_R |= 0x80;// Alternating

GPIO\_PORTE\_DEN\_R &= ~0x80; // PE7 disable digital IO

GPIO\_PORTE\_DIR\_R &= ~0x80; // Setting the direction

GPIO\_PORTE\_AMSEL\_R |= 0x80; // PE7 enable analog function

//PORTJ

GPIO\_PORTJ\_DEN\_R = 0xFF;

GPIO\_PORTJ\_DIR\_R = 0xFF;

// ADC0\_SS0 configuration

ADC0\_ACTSS\_R &= ~0x0F; // disable all 4 sequencers of ADC0

ADC0\_SSMUX0\_R |= 0x00000000; // sequencer 0, channel AIN2,AIN0,AIN1

//ADC0\_SSCTL0\_R |= 0x00000200; // END2 set, sequence length = 3

ADC0\_ACTSS\_R |=0x01;

//Timer Configuration

SYSCTL\_RCGC1\_R |= (1<<16);

i++;

TIMER0\_CTL\_R &= ~0x0001; // disable Timer 0

TIMER0\_CFG\_R = 0x04; // 2 x 16-bit mode

TIMER0\_TAMR\_R = 0x22; // periodic mode + match enable

TIMER0\_TAPR\_R =7 ; // pre-scaler PR= ceil((16M/2^16)\*0.03) -1

}

void displayValue()

{

int total, temp1, dig1,dig2;

while(ADC0\_SSFSTAT0\_R & (1<<8));

total=(int)(ADC0\_SSFIFO0\_R \*5000/1023);

GPIO\_PORTE\_DATA\_R= 0x01;

temp1=total % 100;

dig1= temp1%10;

dig2=(int)(temp1/10);

GPIO\_PORTJ\_DATA\_R= (int)((dig2<<4)|dig1);

total/= 100;

GPIO\_PORTE\_DATA\_R= 0x02;

wait(30);

dig1= total%10;

dig2=(int)(total/10);

GPIO\_PORTJ\_DATA\_R=(int)((dig2<<4)|dig1);

}

void main(void)

{

configSys();

while(1)

{

GPIO\_PORTE\_DATA\_R=0x04;

GPIO\_PORTE\_DATA\_R=0x00;

displayValue();

}

}