

Devices in use :

-Name of the new compiler

-Microcontroller a LM3S9B92 Stellaris

-voltage source of 5v

-Terminal

-control In/out put board

-Keypad

Introduction :

Observation:

Schematic and Nassi schnider diagram:

**Conclusion :**

In this lab we learned how to program a microcontroller, as well as connection between different devices which we used during this lab like microcontroller, I/O board, terminal and keypad.

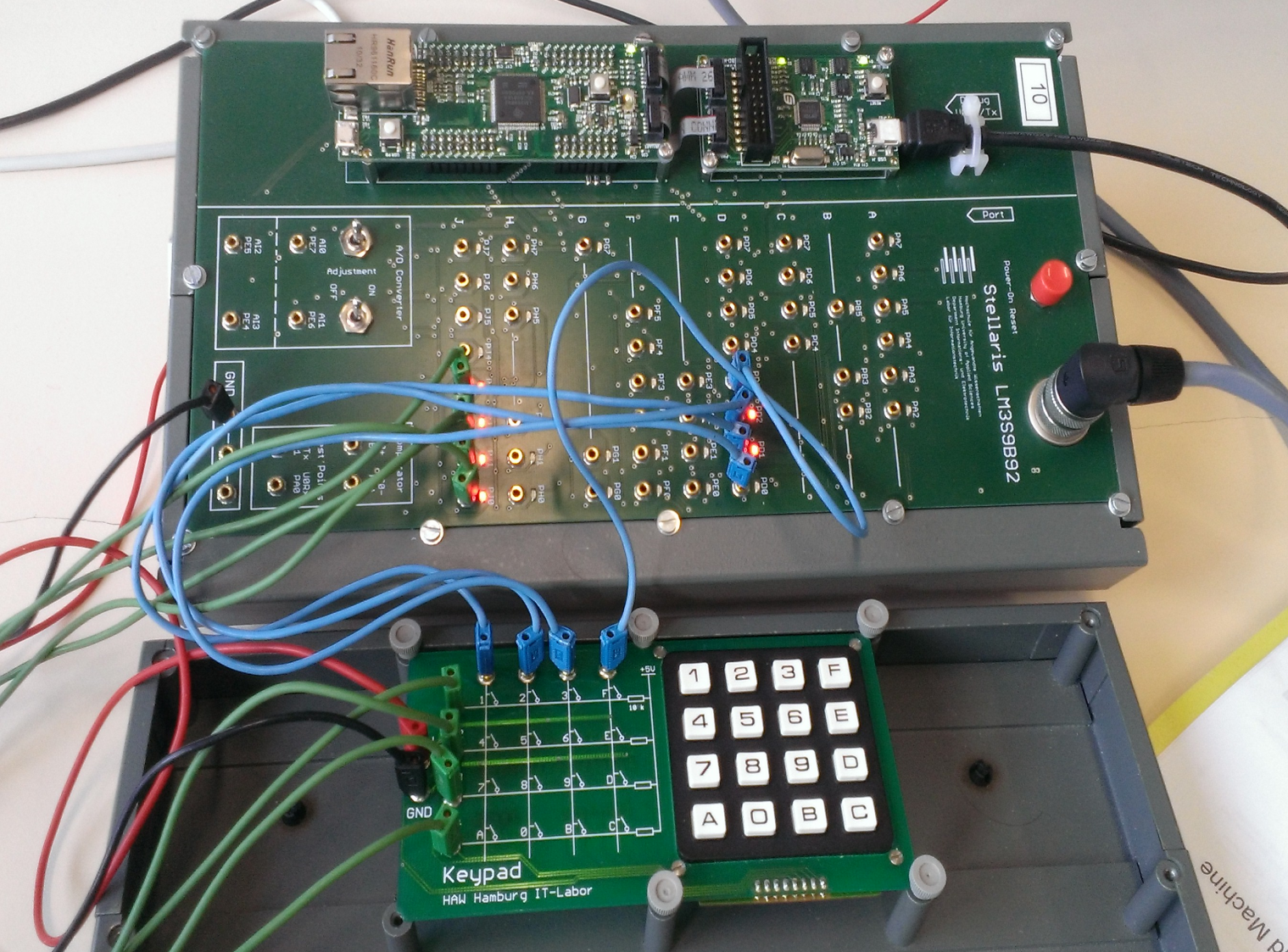
Moreover, by working with new compiler???, we had a new experience in programming with new compiler. We learned how to handle different errors which might be either in our program(solved by debug the program)or connection of our device(solve by close look to our connections and ports).

Finally during the test of our program we saw the major effect of wait loop to synchronize input and output.

**Lab 1 : Keyboard Encoder**

The purpose of the lab task was to program a LM3S9B92 microcontroller and display the result on the terminal using the printf() function. Here in the following setup if no key is pressed Y1 to Y4 are “High” and if any key if X4 is set to low (as programmed) and key “D” is pressed, then Y3 changes to low, Now the key “E” is encoded by machine readable language and we used printf() to display “D” on the terminal of code compressor studio.

Below we have included the picture of our setup.



Fig(1) Hardware setup keypad connected via wire to LM3S9B92 microcontroller

We set up the keypad to be “high” with 5 volt at initial stage. As the key is pressed it detects by probing columns one by one with “low” and look if this causes one of the rows to output “low”. First of all the microcontroller is configured by writing suitable values and initializing data registers.

Then probing is achieved by setting a column to “high” through Port 4 data registers. Input is detected if one of the bits in Data registers becomes “low”. Afterwards the input is evaluated depending on which bit of Port 4 has become “low” and the corresponding character is printed. For this reason 4 bits of port 4 of microcontroller was set for input and 4 bit of the port 8 for the input, which was connected to columns and rows of the keyboard respectively.

Configure the micrcontroller

probe columns Input detected ?

probe next column

Print input

Interprete Input

Fig(2) Software modeling and implementation flow chart

KEYPAD PRESSED ?

Repeat 4 times for coulombs of keypad

Set port D to different input combinations (0x07,0x0D,0x0B,0x0E)

Repeat 4 times for rows of keypad

True False

Print the character. Repeat the loop.

IF not (Port j and row)=0

Fig(3) Nassi –Shneiderman- Diagram of software implementation

Conclusion :

The source code in c programming language for the following Key Encoder is given below.

#define lm3s9b92

#include "lm3s9b92.h"

#include "stdio.h"

#include”stdlib.h”

Void waitloop (void)

{

Long time ;

for(time=0;time<100;time++);

for(time=0;time<10000;time++);

}

void main(void) {

SYSCTL\_RCGC2\_R = SYSCTL\_RCGC2\_R | ((1<<3)|(1<<8));

GPIO\_PORTD\_DEN\_R=0x0F;//enable bits 0 to 3 of Port D

GPIO\_PORTD\_DIR\_R=0x0F;//set bits 0 to 3 of Port D as outputs

GPIO\_PORTJ\_DEN\_R=0x0F;//enable bits 0 to 3 of Port J

GPIO\_PORTJ\_DIR\_R=0x00;//set bits 0 to 3 of Port J as inputs

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

//create the array with all the characters of the keypad:

char ar[4][4]={ {'1','2','3','F'},

{'4','5','6','E'},

{'7','8','9','D'},

{'A','0','B','C'},

};

int i,j; // i is the counter for rows; j is the counter for columns

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

GPIO\_PORTD\_DATA\_R=0x0F; //set bits 0 to 3 of output Port D to "1" so that later we can check when one of them will become "0"

while(1)

{

for(j=0;j<4;j++)//loop through the columns

{

GPIO\_PORTD\_DATA\_R=0x0F^(1u<<j);//set bit j of Port D to 0 (1u is 1)

waitloop(); // make ouput as fast as input without repetition

for(i=0;i<4;i++)//loop through the rows

{

if(!(GPIO\_PORTJ\_DATA\_R & (1u << i)))//if bit i of Port J is 0, the expression will evaluate to true

{

printf("%c\n",ar[i][j]); // print the character in the terminal

}

}

}

}

}