**Project 5 – CECS 262 (Extra Credit)**

**Serial Port Programming and Memory Access**

Student’s Name: Len Quach

Due Date: 5/8/2020

The project is to learn how to program 8051 serial port and access the 8051’s different memory areas. We will need to write a program that excesses the 8051 Internal RAM, Expanded RAM as an optional, and Flash ROM using the user interface.

The program will be starting to display the main menu at first:

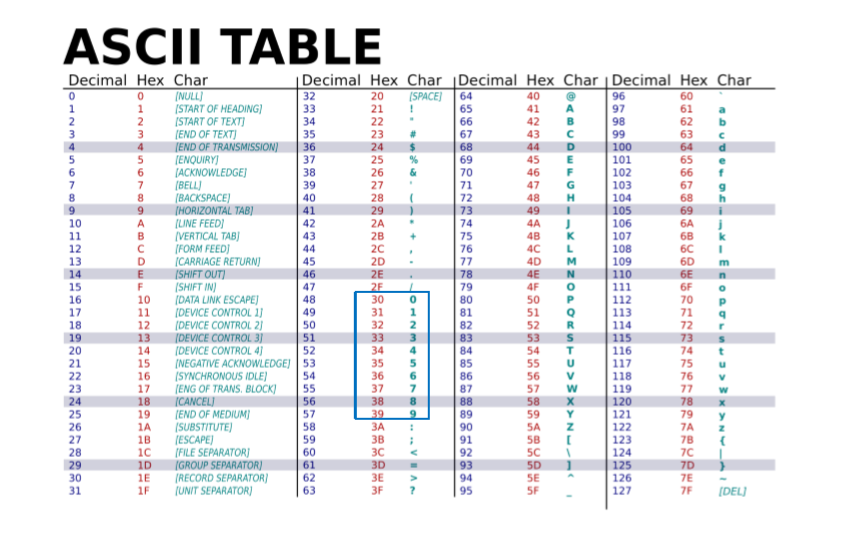
“This program allows you access different memory areas in 8051.

Please choose a memory area:”

So we will have 3 options displayed and ask the user to choose one of them: Internal RAM, Flash ROM, and Expanded RAM. If the user selects the Flash ROM option, we will ask them to enter the starting and ending address”. However, if the user selects either Internal RAM or Expanded RAM, then we need to ask the user if they want to read or write first (enter R for read and W for write) and after that, ask for enter the starting and ending address later. The program will display the contents in the specified address to read the operation, and show the user to enter data in hex for the write operation. We can confirm the written data by displaying it after writing the operation. There will be spaces between the data showing.

For example, in case if the user chooses option1 to read 4 bytes from Internal RAM, enter the starting address is 70 in Hex and ending address is 73H, then the memory in internal RAM locations of 70H to 73H or 70, 71, 72, 73 will be in order of 0A, F5, 6A, 67.

“Define” is a global variable, essentially. When we #define something at the beginning of a program and it will always equal whatever the second thing we past it in that line.



Here is the code:

//===============================================================

#include <reg51.h>

#include <stdio.h>

#define CR 0x0D //define carriage return as CR

#define LF 0x0A //define line feed as LF

#define uchar unsigned char //define unsigned character as uchar

#define uint unsigned int // define unsigned integer as uint

receive(void); //receive text function

void send(unsigned char \*s); //send text

void write\_mem(uint i, uint mode); // write to RAM

void read\_mem(uchar mem\_value, uchar\* value\_str); // read from ROM

void main(void)

{

unsigned char a,c,s,e,t,d;

unsigned int i;

while(1) { //repeat step 1

send("This program allows you access to different memory areas in 8051.\n");

send("Please choose a memory area:\n");

send("1. Internal RAM \n2. Flash ROM \n3. Expanded RAM \n");

a = receive();

if (a == 'o') {

send("Do you want to read or write?(Enter R for read and W for write):\n");

}

c = receive();

if (c == 'R')

{

send("Internal RAM\n");

send("Please enter starting address:");

s = receive();

if (s) {

send('s');

//read\_mem(s);

//for(i=0;i<=3500;i++);

}

send("\nPlease enter ending address:");

e = receive();

if (e) {

send('e');

//read\_mem(e);

//for(i=0;i<=3500;i++);

}

send("\nThe contents in internal RAM locations 70H - 73H are: ");

}

if (c == 'W') {

send("Expanded RAM\n");

//for(i=0;i<=3500;i++);

send("Please enter starting address:");

t = receive();

if (t) {

send('t');

//read\_mem(t);

//for(i=0;i<=3500;i++);

}

send("\nPlease enter ending address:");

d = receive();

if (d) {

send('d');

//read\_mem(d);

//for(i=0;i<=3500;i++);

}

send("\nExpanded RAM locations F2H - F4H are updated with: ");

}

}

for(i=0;i<=3500;i++); //for loop

}

void send(unsigned char \*s)

{

SCON = 0x50; //mode 1, baudrate generating using timer 1

TMOD = 0x20; //Timer 1, mode 2

TH1 = -3;

TR1 = 1; //run the timer

while (\*s) {

SBUF = \*s++; //send the character in the address of variable s being increasing

while (TI == 0);

TI = 0;

}

}

receive(void)

{

unsigned char rb;

while (RI == 0);

RI = 0;

rb = SBUF; // receive data is stored into the rb variable

return rb;

}

void read\_mem(uchar mem\_value, uchar\* value\_str) //read memory

{

value\_str[0] = (mem\_value&0xF0)>>4; //shift to right 4

value\_str[0] = value\_str[0]>9?

value\_str[0]-10+'A':value\_str[0]+'0';

value\_str[1] = mem\_value&0x0F;

value\_str[1] = value\_str[1]>9?

value\_str[1]-10+'A':value\_str[1]+'0';

value\_str[2] = ' '; //space

value\_str[3] = '\0'; //null character

}

//======================================================

This serial port programming and memory access project and its lectures helps me to know how the serial communication works with the COM port of the PC, and know to program for serial port. For some reason, I was not be able to connect the serial port using HyperTerminal to display the messages and gave me a hard time on this project. However, supposedly Keil has an in-buit display for serial data, so I used the serial window instead of HyperTerminal or Putty, and it still works pretty well.

<https://www.youtube.com/watch?v=HgJ9_hPpl-E&feature=youtu.be>