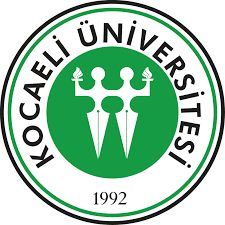
**KOCAELI UNIVERSITY**



**Department of**

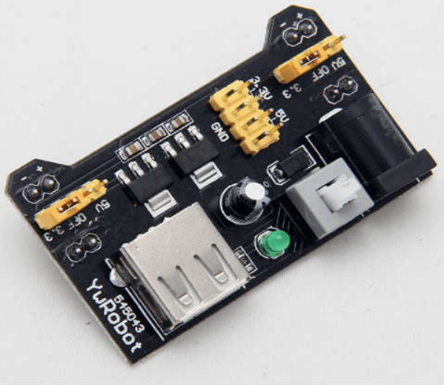
**Electronics & Communication**

**Engineering**

**Microprocessor Lesson’s Project’s Report**

Name: Bahar Ateş

My components

1. Voltage Supply unit

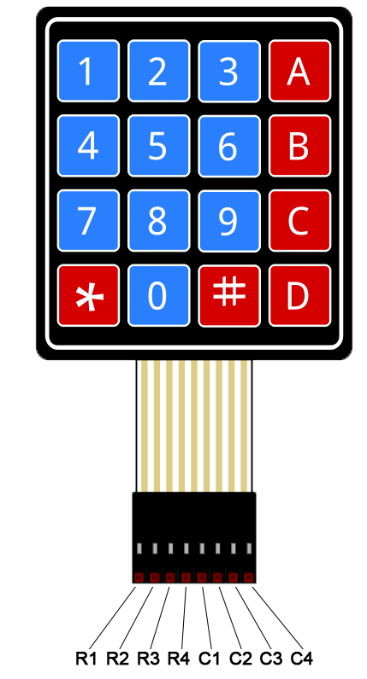
MB102 supporting 5 V or 3.3 V output.

Input voltage: DC 6.5-12 V.

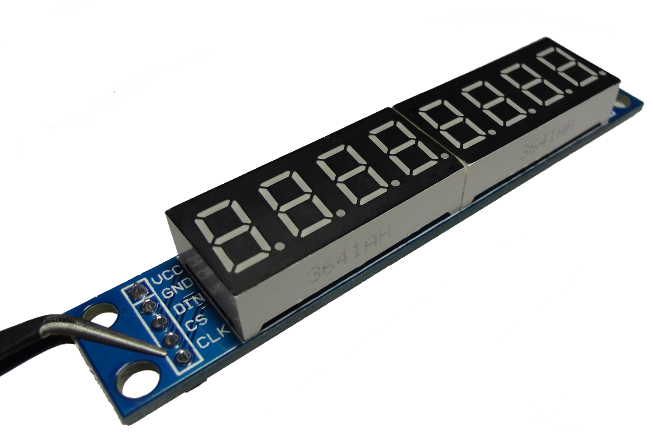
Output voltage: 3.3V/5V

Maximum output current: 700 mA.

Attach the male jumpers to the female jumpers for 3.3 volt. After than one terminal should be connected to the ground and other terminal is the input voltage 3.3 volt.

2)Keypad

Matrix keypads use a combination of four rows and four columns to provide button states to the host device, typically a microcontroller. Underneath each key is a pushbutton, with one end connected to one row, and the other end connected to one column. In order for the microcontroller to determine which button is pressed, it first needs to pull each of the four columns (pins 1-4) either low one at a time, and then poll the states of the four rows (pins 5-8). Depending on the states of the columns, the microcontroller can tell which button is pressed.

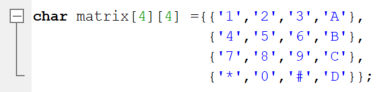
3)Display: MAX7219

The MAX7219 is compact, serial input/output common-cathode display driver that interface microprocessors (µPs) to 7-segment numeric LED displays of up to 8 digits, on-chip is a BCD code-B decoder, multiplex scan circuitry, segment and digit drivers, The MAX7219 is compatible with SPI™.

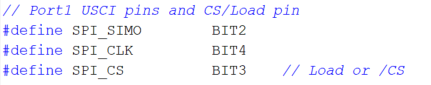
Code Explanation

I am using “io430.h” for using msp430 device.

“stdint.h” header defines a set of integral type aliases with specific width requirements, along with macros specifying their limits and macro functions to create values of these types.

This matrix corresponds my keypad’s buttons.

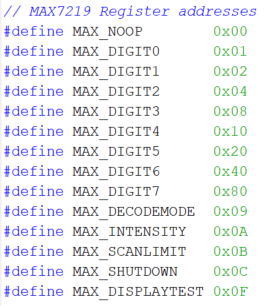
Keypad has 4 rows and 4 columns. And I defined each element of the array .

These lines define the msp430 and display's serial communication pins.

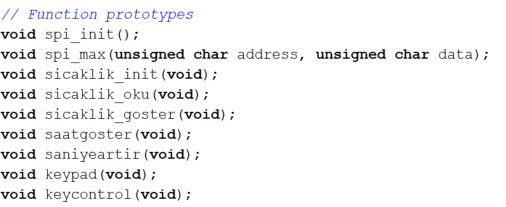
SIMO is Serial-Data Input. Data is loaded into the internal 16-bit shift register on CLK’s rising edge.

CLK is Serial-Clock Input. 10MHz maximum rate. On CLK’s rising edge, data is shifted into the internal shift register.

CS is Chip-Select Input. Serial data is loaded into the shift register while CS is low. The last 16 bits of serial data are latched on CS’s rising edge.



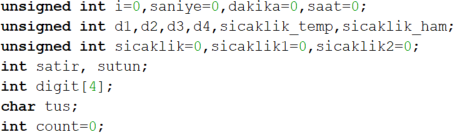
These lines are 14 addressable digit and control registers. The digit registers are realized with an on-chip, 8x8 dual-port SRAM. They are addressed directly so that individual digits can be updated .The control registers consist of decode mode, display intensity, scan limit (number of scanned digits), shutdown, and display test (all LEDs on).

Function prototype tells compiler about number of parameters function takes, data-types of parameters and return type of function. So I have to define all my functions.

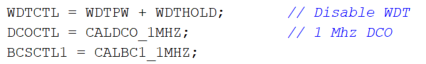
Each function will be further elaborated in this document.

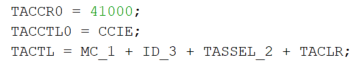


This array corresponds to numbers : 0 1 2 3 4 5 6 7 8 9 .

 These are the variables I use in functions. Each variable will be further elaborated in related function. I made them global variables because I want to use them in many places.

First, I want to explain main function. This function has no return, so I made it void.

First of all, We have to set internal digital clock oscillator.It is an RC oscillator, which can be controlled as software, starting to operate within 1us.

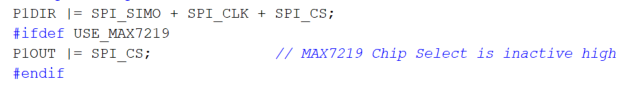


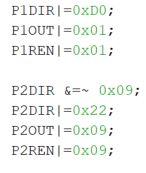
MC\_1: The timer continuously counts from zero to CCR. CCR is 41000.

The CCIE bit in the TACCTL0 register is Logic1. In this way, the use of interrupt for timer is activated.

The TASSEL bits are set to TASSEL\_2. In other words, the clock pulse coming to the timer will be provided by SMCLK.

TACLR means clear timer.

This definition for set up Max7219 pins.

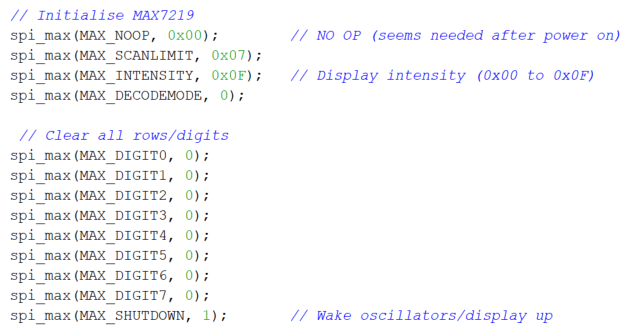


Port 1’s P1.0 P1.1 bit is input; Port 1’s P1.4 , P1.6, P1.7 bits are output.

Port 2’s P2.0, P2.3 bits are input; Port 2’s P2.1 , P2.5 bits are output.

I initialize temperature sensor .

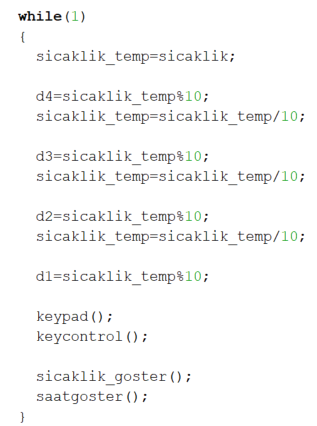
I initialize USCI in SPI mode.

 I made SR register's GIE bit is enabled.

Decode mode is off. Display intensity is maximum. Scan limit (number of scanned digits) includes all .

I sent 0 to all segments.

Shutdown is off. So Max7219’s oscillators are wake up. Accordingly the displays start to work.

In main function, I wanted to run some codes forever. So, I put them in while(1). While(1) means,it always true, do this forever.

First, I made a temporary variable assignment to the temperature value because I did not want to lose the value when measuring the temperature in the back.

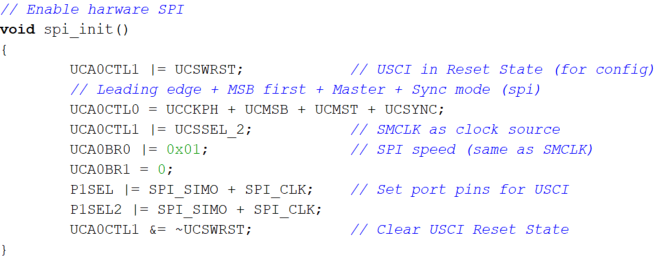
After this, I separated temperature to digits to display.For example, my temperature value is 2645, d4 stores 5, d3 stores 4, d2 stores 6, d1 stores 2.

After the separation, my code goes to keypad scan function.

After the scanning, I have to check which button is pressed. And what will i do.

And for the show temperature and time in display, code has to go sicaklik\_goster function and saatgoster function.

Let’s look my functions.



UCSWRST bit is used to put the USCI module in reset state .It is recommended to put the USCI module in reset state before making any changes to the registers.

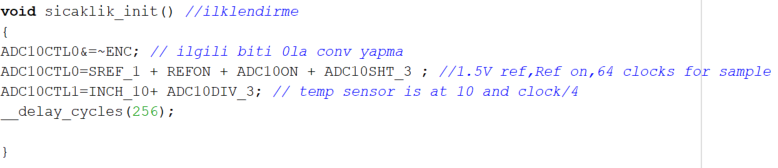
UCSYNC, selecting between asynchronous(UART) and synchronous modes(SPI).

UCMSB controls the direction of receive and transmit shift register(here LSB first).

The polarity and phase are controlled by UCCKPH.

Master mode is selected with UCMST.

The frequency is then divided by the value in the UCB0BR1:UCB0BR0 registers.



Enc bit is used to enable adc conversion.

Sref bit is used to choose referance voltage.

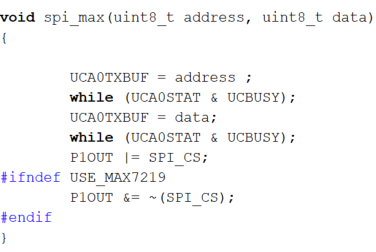
Refon bit is activate reference voltage.

ADC10ON bit is activate ADC10.

ADC10SHT is determine the sampling time.

INCH is shows which analog input is selected.

ADC10DIV is determine frequency divider.

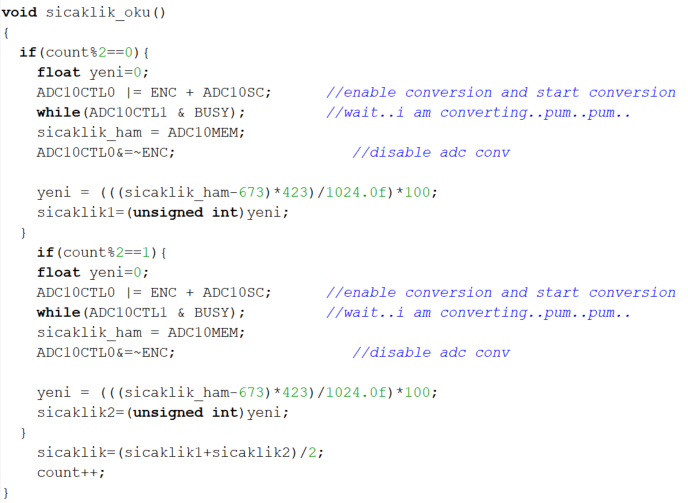
In this function, I am sending adress and data with SPI.

A transfer begins when a value is written to the transmit buffer TXBUF. The clock is

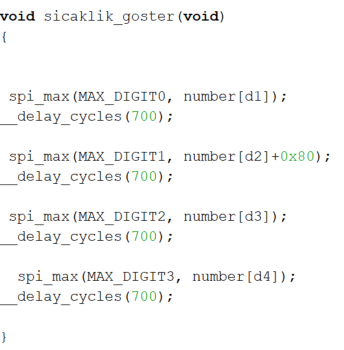
started in a master, while a slave waits for a clock signal on its CLK input, and the flag

UCBUSY is raised to show that the module is busy.

The #ifndef directive checks the reversal of the condition controlled by #ifdef. If the identifier is not defined (or if the definition is removed with #undef), the condition changes to true (nonzero). Otherwise, the condition is false (0).



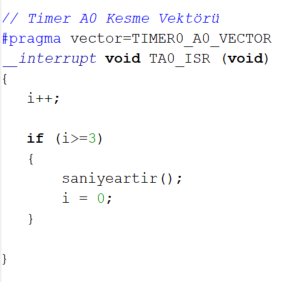
I placed 2 conditional functions into this function. I increased a variable named Count when I entered this function. The code entered one of the functions according to mode 2 of count. If count%2=0 , the code enters the first function. . If count%2≠0 , the code enters the second function. Basically they both measure temperature and assign measured temperature two different variables. I wanted to avoid the deviation by taking the average of these variables. But I have not been successful. I know I need to get more of the value in order to avoid deviation.

In this function, I am sending address and data to spi\_max function. I separated temperature to digits in main function.

Now, I am sending them to MAX7219.

I am waiting a little while. Because we can’t display all digits at the same time. We have to change them quickly. This delay for this.

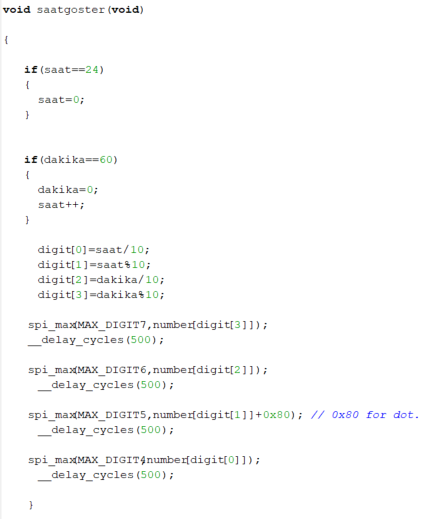
You can see, I am sending external number to digit2. This number means a dot. I want to separate them with a dot.

This function is timer interrupt function. When timer interrupt comes, code entered this function. I increased a variable named i when I entered this function. When i greater then 3, code goes to function named saniyeartir.I reset i to start over.

I increase seconds with this method beacuse other way times did not match. Now I can compare with real time.

Code came from timer interrupt function. I increased a variable named saniye when I entered this function. And every second, I sent the code to function sicaklik\_oku to read the temperature.

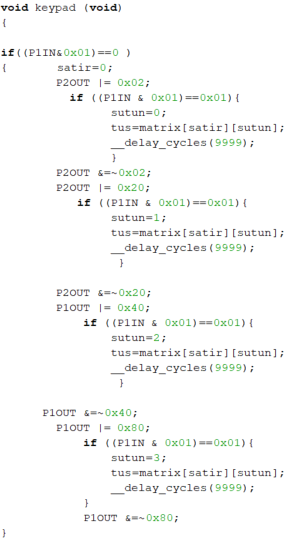
Seconds and minutes can not be greater than 59 and hours can not be greater than 23. In this function, It is controlled. If saniye is equal to 60, now I have to increment dakika. If dakika is equal to 60, now I have to increment saat. If saat is equal to 24, now I have to increment dakika. I must reset the saat.

We started the time at the previous function. Now we can display the time. I controlled minute and hour again because I increment time in external place too.

I separated time to digits. Now, I can send them to MAX7219 and I am waiting a for a while to display other digit.

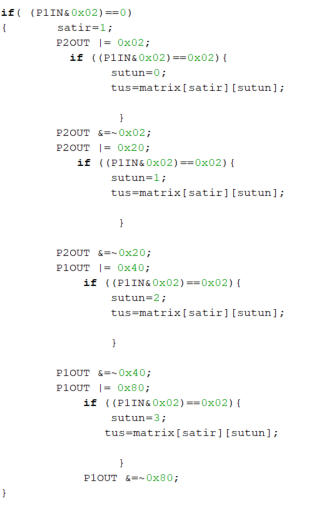
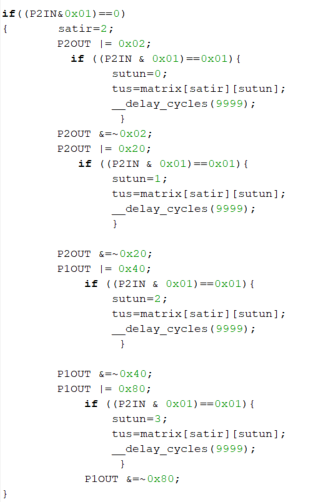
And again I am sending external number to digit5 to separate them with a dot.

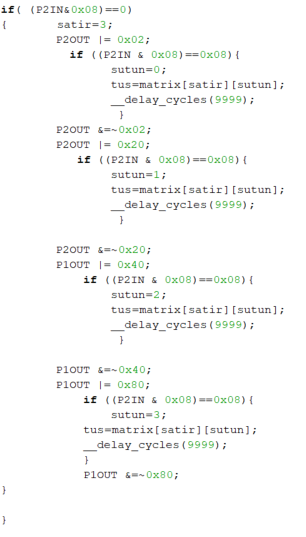
Now, I want to tell how I did scan.

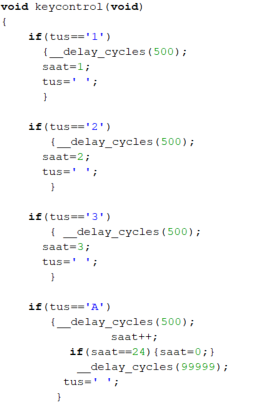
My rows are input, my columns are output.

I thought like button logic. I do and operation to Port 1 with first row.If result is 0, row is 0. Now I have to check which column is. I send 1 to the columns respectively. I'm doing the and operation with the selected row and when it gives 1, the column is written.Previously, I have defined the matrix according to keypad. I can learn the key by sending row and column to matrix.

I repeat this cycle for every row.



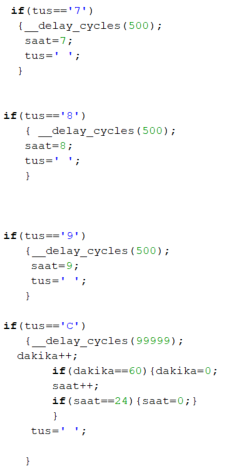
I can’t use row2 because of due to a hardware problem.

Now I write custom codes for each key so that it is done when pressed. This is for first row. For numbers, I set the time to which key pressed.

I increased the hour for A. And I controlled hour again.

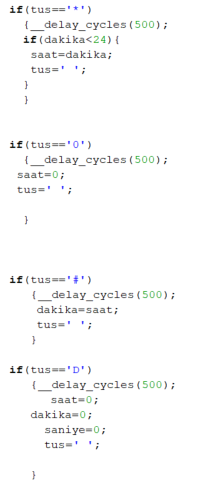
I set the value blank for the pressed key to doesn't repeat the event.

I wait a little more in the minute and hour increments for the keystroke pass.



This is for third row. For numbers, I set the time to which key pressed.

I increased the minute for C. And I controlled minute an hour again.

I set the hour to minute for \*. but I can do it if it's smaller than 24.

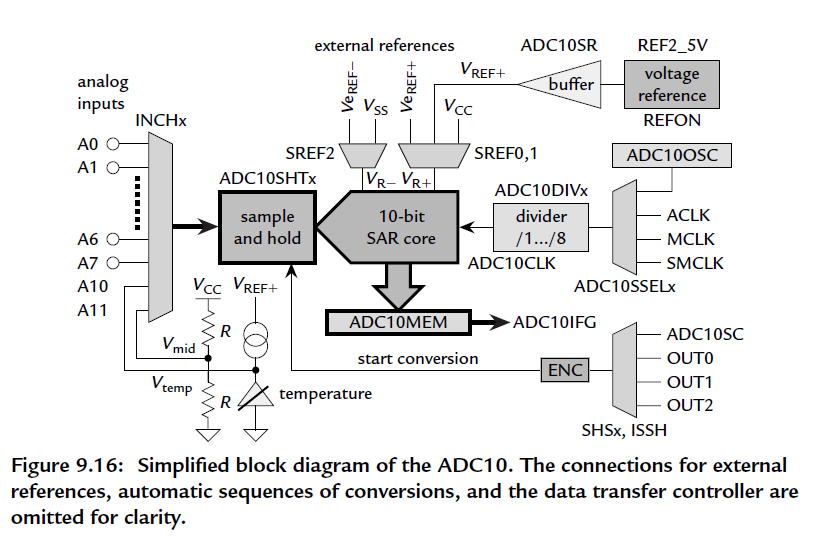
For 0, I set hour to 0.

I set the minute to hour for #.

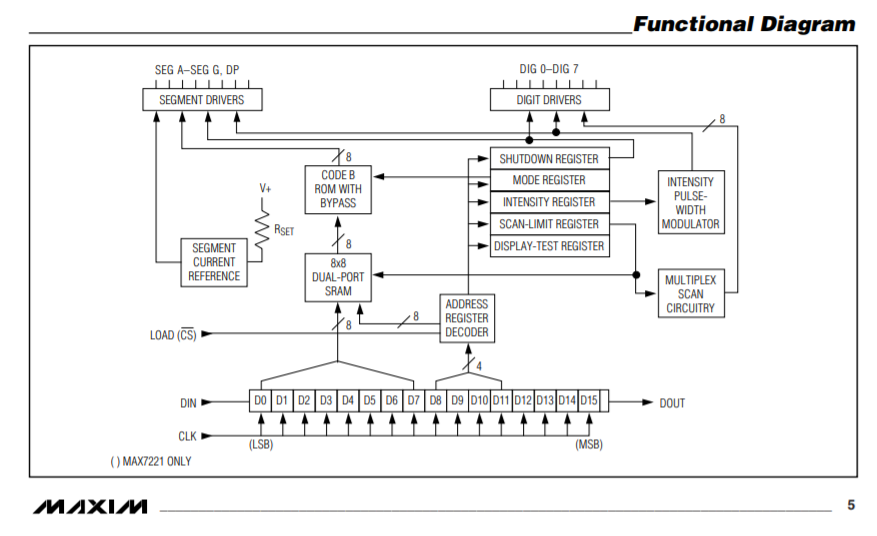
I reset the time for D.

Additional informations

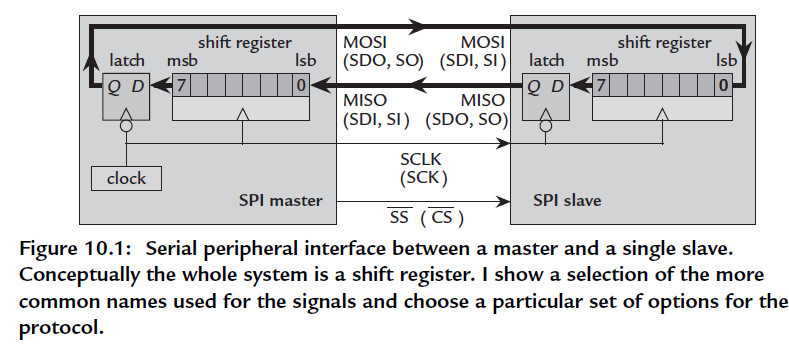
1. Block diagram of the ADC110



1. Max7219 Functional Diagram



1. SPI Interface



1. USCI block diagram in SPI mode

