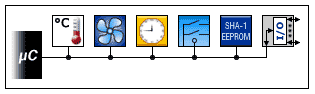
**MSP430 Uyg.33 – DS1990a iBUTTON Uygulaması**

Ferudun GÖKCEGÖZ, 30 Ağustos 2011, Salı



Merhaba arkadaşlar. MSP430 uygulamalarına kaldığımız yerden devam ediyoruz. Bu dersimizde iButton olarak bilinen, maxim(dallas semi) firmasına ait DS1990a cihazını msp430 ile kullanacağız. Yandaki fotoğraftada görüldüğü gibi ibutton, bildiğimiz akbildir. Ve maxim firması tarafından üretilmiş birçok türevi bulunmaktadır. DS1990a modeli ise en basit olanıdır. Biz bu uygulamada sadece ds1990a yı kullanacağız. Ayrıca OneWire protokolünden de bahsedeceğiz. Bir sonraki dersimizde ise yine OneWire protokolüyle haberleşen DS18B20 sıcaklık sensörü ve termostat entegresini kullanarak bir uygulama yapacağız.  İlk olarak OneWire yani tek hat üzerinden haberleşme protokolünden bahsedelim.



1wire protokolü maxim – dallas firmasının kendi ürünlerinde standartlaştırdığı bir haberleşme protokolüdür. Tek hat üzerinden haberleşmeyi sağlar. Daha önceden RS232, SPI, IIC gibi haberleşme protokollerini duymuşuzdur. Bu protokolde diğerleri gibi haberleşme arayüzü sağlar fakat diğer firmalar tarafından tam olarak benimsenmemiştir. Ama oldukça kullanım kolaylığı sağlar. Ve bu arayüz üzerinden çift yönlü haberleşme sağlanabilir. Ayrıca besleme hattına gerek duymaz. Enerjisini hattan alabilme yeteneğine sahiptir. “Parasite Power” ismindeki güç sistemiyle hattaki enerjiyi dahili bir kapasite üzerinde toplayarak enerjisini oradan sağlar. DS18B20 gibi yine bu protokol ile haberleşen farklı ürünlerde ayrıca besleme girişide bulunmaktadır. Bu şekilde enerjisini hat üzerinden değilde, besleme pininden de sağlayabilir. Bu durum uygulamalarda bize oldukça esneklik sağlar. Ayrıca yine maxim firmasının ürettiği IIC den OneWire protokolüne dönüştürme entegreleride mevcuttur. Bu şekildede kullanılabilir.

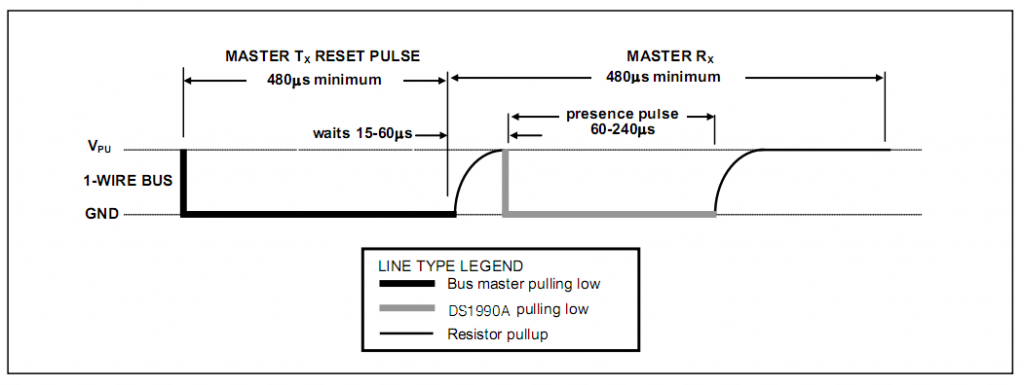
Yukarıdaki şekilde de görüldüğü tek hat üzerinde birden çok OneWire cihaz bulunmaktadır. Bu cihazlar üzerinde ürünün üretim aşamasında dahili ROM Hafızasına kaydedilen 64 bitlik bir ROM Code u bulunmaktadır. Bu ROM Code her ürün için farklıdır. Bu şekilde hattaki birden çok OneWire cihazı tanıyabilme veya eşleştirme mümkün kılınmıştır. One Wire cihazlar hakkında daha detaylı bilgiler için maxim firmasının [**web sitesi**](http://www.maxim-ic.com/auto_info.cfm) incelenebilir.

OneWire protokolüyle veri alışverişi yapabilmek için 3 adet temel fonksiyon vardır. Bunlardan biri hattı resetlemek için, diğer ikisi ise 1 bit veri okumak ve 1 bit veri yazmaktır. Bu üç fonksiyon OneWire protokolüyle haberleşebilmek için kullanılan temel fonksiyonlardır. 1 byte veri okumak ve 1 byte veri yazmak için ise, 8 kere bit yazma veya okuma fonksiyonları tekrarlanarak gerçekleştirilir. Yani OneWire haberleşme ile ilgili bir kütüphane oluşturmak istesek toplamda 5 adet fonksiyonumuz olacaktır.

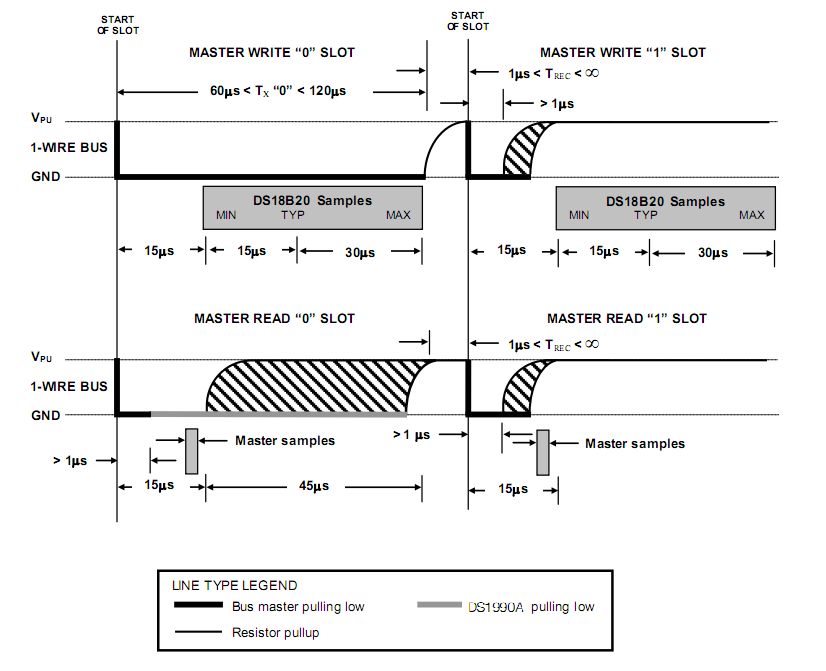
* Hattı resetleme fonk.
* 1 bit veri yazma fonk.
* 1bit veri okuma fonk.
* 1 byte veri yazma fonk.
* 1 byte veri okuma fonk.

Uygulamaya ait verilecek olan yazılımda bu bahsettiğimiz fonksiyonlara ait alt programları kütüphanelerde bulabilirsiniz.

Gelelim bu 3 temel fonksiyon için timing diyagramlarına…

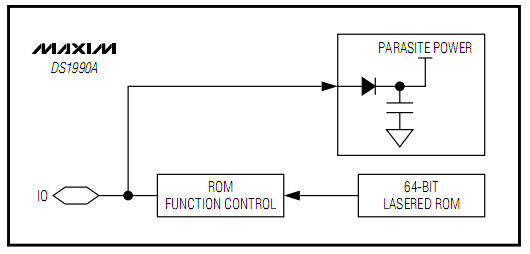


Kısaca yukarıda timing diyagramından reset pulse ı açıklayacak olursak, hat ilk önce master tarafından sıfıra çekilir ve belirtilen süre sonunda hat boşta bırakılarak pull up direncinin etkisiye lojik  1 çekilmesi beklenir. Daha sonra OneWire cihaz tarafından hat sıfıra çekiliyorsa, hatta bir cihaz var demektir. Eğer pull up direnciyle hat lojik1 çekildikten sonra hala lojik1 te kalıyors hatta cihaz yok demektir. Bu zamandan sonra yine belirtilen süre kadar beklenilir ve pulse tamamlanır. Bu fonksiyon aslında hattaki slave cihaz için bir initialization(başlatma) komutudur.

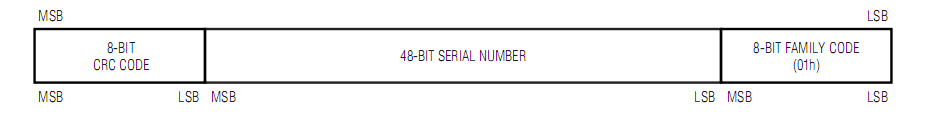
Yukarıda ise master tarafından lojik1 gönderme, lojik0 gönderme, lojik1 okuma ve lojik0 yazma için gerekli timing diyagramları verilmiştir. Detaylı şekilde incelemek isteyenler bakabilirler. İnternette bu timingler hakkında birçok yazılmış makale, yapılmış uygulama ve detaylı timing diyagramları mevcuttur. O yüzden bu konu üzerinde daha fazla durmadan DS1990a iButton cihazından bahsetmek istiyorum.



Birçoğumuzun istanbuldaki toplu taşıma araçlarında kullandığımız, şimdilerde ise tarih olmaya yüz tutmuş bir cihaz ile karşı karşıkarşıyayız. Her ne kadar toplu taşıma araçlarında yeni uygulamalara geçilsede iButton birçok yerde güvenle kullanılmaktadır. DS1990a 2.8V tan 6V a kadar besleme aralığında, -40 ~+85 C derece arasında sıcaklıkta çalışabilecek bir cihazdır. İki tür kılıf olarak üretilmektedir. F3 ve F5 olmak üzere… F3 modelinin kalınlığı 3.10mm, F5 modelinin ise 5.89 mm dir. Yazının başında bahsettiğimiz gibi üretim esnasında dahili ROM hafızasına kayıtlı 64 bitlik ROM Code u bulunmaktadır. Rom hafızadan ayrı olarak bir Ram hafızası bulunmamaktadır. Tabi daha üst model iButtonların Ram hafızalı olanları bulunmaktadır.  16.3 kbps hızında haberleşebilmektedir. Aşağıda ürüne ait blok diyagram görülmektedir.



One Wire cihazlarının tümünde 64 bitlik bir Rom code bulunmaktadır. Bu kodun en düşük 8 biti family code(aile kodu) dur. DS1990a için aile kodu : 0×01 dir. Bu 8 bitten sonraki 48 bit ise cihaza ait seri no dur. Ve her cihazda farklıdır. Son 8 bit ise daha önceki 56 bitin CRC algoritmasından geçirilerek elde edilmiş 8 bitlik CRC kodudur. Aşağıda toplam 64 bitlik datayı tablo halinde görülmektedir.



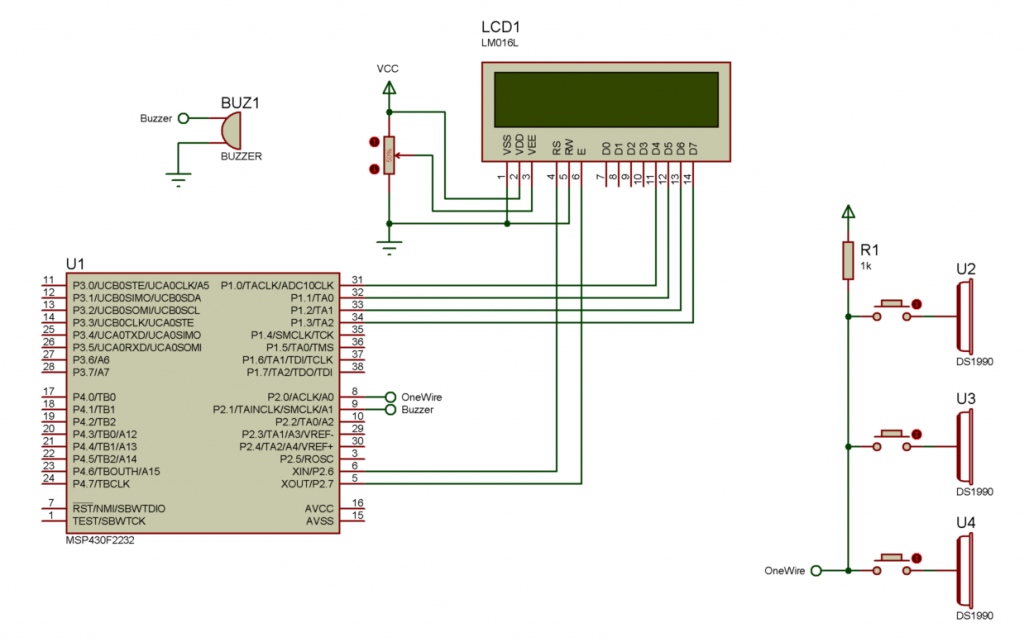
Eğer hatta bağlı birden fazla cihaz var ise cihaza ait seri no okunarak cihazlar tanımlanabilir. Bizim yapacağımız uygulamada ise 3 adet iButton kullanılmıştır. Seri numaraları kıyaslanarak basılan iButton un kime ait olduğu saptanmıştır.

CRC algoritmasından bahsedecek olursak, CRC bir hata kontrol algoritmasıdır. 8 ve 16 bitlik türleri bulunmaktadır. Burada kullanılan 8 bitlik olanlardandır. 8 bitlik olanlardan diyorum çünkü farklı çeşitlerde 8 bitlik CRC algoritmaları kullanılmaktadır. CRC kodu hattan okunan 56 bitlik(7 byte) family code + serial number datasının doğruluğunu test etmek için kullanılabilir. Yani toplamda okunan 8 byte lık datanın ilk 7 byte ı CRC algoritmasından geçirilir ve elde edilen 1 byte lık CRC kodu, okunan CRC kodu ile karşılaştırılarak, hattan okunan verinin doğruluğu denetlenebilir.

OneWire cihazlarında cihaza ait birer byte lık fonksiyon kodları bulunmaktadır. DS1990a ya ait fonksiyon kodlarınıda kısaca yazacak olursak;

* **READ ROM = 0×33**
* **SEARCH ROM = 0xF0**
* **MATCH ROM = 0×55**
* **SKIP ROM = 0xCC**

Bizim yapacağımız uygulamaya ait devre şemasını vererek uygulamamıza geçebiliriz.



Devre şemasındada görüldüğü gibi OneWire hattına bağlı 3 adet iButton bulunmakta. Bu iButtonların seri numaraları ayarlanabilir. Bende 3 iButton un seri numaralarını birbirinden farklı olacak şekilde ayarladım.

Farzedelimki, en üstteki iButton AHMET e ait akbil olsun. Ortadaki MEHMETe ait akbil olsun. En alttaki ise AYŞE ye ait akbil olsun. Yazılım içerisinde iButton ların sol kısmındaki switch lere bastığımızda aslında akbili dokunduruyorumuşuz gibi düşünebiliriz. Hiçbir akbil dokundurulmamış ise hatta cihaz olmadığı LCD ekran üzerinden gösterilen mesajla bize bildirilecektir.

Uygulamamıza ait yazılımıda verecek olursak;

|  |  |
| --- | --- |
| **[OneWire.c](http://www.mcu-turkey.com/msp430-uyg-33/" \l "codesyntax_1" \o "Click to show/hide code block)** | **[http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/themes/default/images/code.png](http://www.mcu-turkey.com/msp430-uyg-33/#codesyntax_1)** **[http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/themes/default/images/printer.png](http://www.mcu-turkey.com/msp430-uyg-33/#codesyntax_1)** **[http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/themes/default/images/info.gif](http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/About.html)** |

#include "io430.h"

#include "in430.h"

#include "OneWire.h"

unsigned char Ow\_Reset(void)

{

unsigned char retVal;

OW\_WRITE\_PIN = LOW;

OW\_PIN\_DIRECTION = OUTPUT;

\_\_delay\_cycles(500);

OW\_PIN\_DIRECTION = INPUT;

\_\_delay\_cycles(70);

retVal = OW\_READ\_PIN;

\_\_delay\_cycles(500);

return retVal;

}

unsigned char ReadBit(void)

{

unsigned char bitDATA;

OW\_WRITE\_PIN = LOW;

OW\_PIN\_DIRECTION = OUTPUT;

\_\_delay\_cycles(5);

OW\_PIN\_DIRECTION = INPUT;

\_\_delay\_cycles(10);

if(OW\_READ\_PIN) bitDATA = 1;

else bitDATA = 0;

\_\_delay\_cycles(60);

return bitDATA;

}

void WriteBit(unsigned char bDATA)

{

if(bDATA)

{

OW\_PIN\_DIRECTION = OUTPUT;

OW\_WRITE\_PIN = LOW;

\_\_delay\_cycles(5);

OW\_PIN\_DIRECTION = INPUT;

\_\_delay\_cycles(60);

}

else

{

OW\_PIN\_DIRECTION = OUTPUT;

OW\_WRITE\_PIN = LOW;

\_\_delay\_cycles(60);

OW\_PIN\_DIRECTION = INPUT;

}

}

unsigned char Ow\_ReadByte(void)

{

unsigned char i,byteData=0, bits=1;

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

{

if(ReadBit()==1) byteData |= bits;

bits<<=1;

}

return byteData;

}

void Ow\_WriteByte(unsigned char byteData)

{

unsigned char bits=1, i;

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

{

if (byteData & bits) WriteBit(1);

else WriteBit(0);

bits<<=1;

}

}

|  |  |
| --- | --- |
| **[OneWire.h](http://www.mcu-turkey.com/msp430-uyg-33/" \l "codesyntax_2" \o "Click to show/hide code block)** | **[http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/themes/default/images/code.png](http://www.mcu-turkey.com/msp430-uyg-33/#codesyntax_2)** **[http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/themes/default/images/printer.png](http://www.mcu-turkey.com/msp430-uyg-33/#codesyntax_2)** **[http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/themes/default/images/info.gif](http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/About.html)** |

#ifndef ONEWIRE\_H\_

#define ONEWIRE\_H\_

#define OW\_PIN\_DIRECTION P2DIR\_bit.P0

#define OW\_WRITE\_PIN P2OUT\_bit.P0

#define OW\_READ\_PIN P2IN\_bit.P0

#define LOW 0

#define HIGH 1

#define OUTPUT 1

#define INPUT 0

#define SET 1

#define CLEAR 0

**extern** unsigned char Ow\_Reset(void);

**extern** unsigned char Ow\_ReadByte(void);

**extern** void Ow\_WriteByte(unsigned char byteData);

#endif /\*ONEWIRE\_H\_\*/

|  |  |
| --- | --- |
| **[main.c](http://www.mcu-turkey.com/msp430-uyg-33/" \l "codesyntax_3" \o "Click to show/hide code block)** | **[http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/themes/default/images/code.png](http://www.mcu-turkey.com/msp430-uyg-33/#codesyntax_3)** **[http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/themes/default/images/printer.png](http://www.mcu-turkey.com/msp430-uyg-33/#codesyntax_3)** **[http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/themes/default/images/info.gif](http://www.mcu-turkey.com/wp-content/plugins/wp-synhighlight/About.html)** |

#include "io430.h"

#include "in430.h"

#include "lcd\_4bit.h"

#include "OneWire.h"

#define READ\_COMMAND\_DS1990A 0x33

#define BUZZER P2OUT\_bit.P1

unsigned char AHMET[] = {0xB8, 0xC5, 0x2C};

unsigned char MEHMET[] = {0xB8, 0xC5, 0x2D};

unsigned char AYSE[] = {0xB8, 0xC5, 0x2E};

unsigned char Detect\_Slave\_Device(void)

{

if (!Ow\_Reset())return HIGH;

else return LOW;

}

void Beep(void)

{

BUZZER = 1;

\_\_delay\_cycles(50000);

BUZZER = 0;

}

void main(void)

{

unsigned char serial\_number[8];

unsigned char temp;

WDTCTL = WDTPW + WDTHOLD;

DCOCTL=CALDCO\_1MHZ;

BCSCTL1=CALBC1\_1MHZ;

P1SEL = 0x00;

P1OUT = 0x00;

P1DIR = 0xFF;

P2SEL = 0x00;

P2OUT = 0x00;

P2DIR = 0xFF;

lcd\_init();

lcd\_goto(1,1);

lcd\_puts("DS1990a iBUTTON");

\_\_delay\_cycles(1000000);

lcd\_clear();

for(;;)

{

if (!Detect\_Slave\_Device())

{

lcd\_clear();

lcd\_goto(1,1);

lcd\_puts("Hatta Cihaz yok!");

}

else

{

lcd\_clear();

lcd\_goto(1,1);

Ow\_WriteByte (READ\_COMMAND\_DS1990A);

lcd\_goto(1,1);

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

serial\_number[temp] = Ow\_ReadByte();

if((serial\_number[3]==AHMET[0]) && (serial\_number[2]==AHMET[1]) && (serial\_number[1]==AHMET[2]))

{

lcd\_goto(2,1);

lcd\_puts(" AHMET in AKBILI ");

*//Beep();*

}

else if((serial\_number[3]==MEHMET[0]) && (serial\_number[2]==MEHMET[1]) && (serial\_number[1]==MEHMET[2]))

{

lcd\_goto(2,1);

lcd\_puts("MEHMET in AKBILI ");

*//Beep();*

}

else if((serial\_number[3]==AYSE[0]) && (serial\_number[2]==AYSE[1]) && (serial\_number[1]==AYSE[2]))

{

lcd\_goto(2,1);

lcd\_puts(" AYSE nin AKBILI ");

*//Beep();*

}

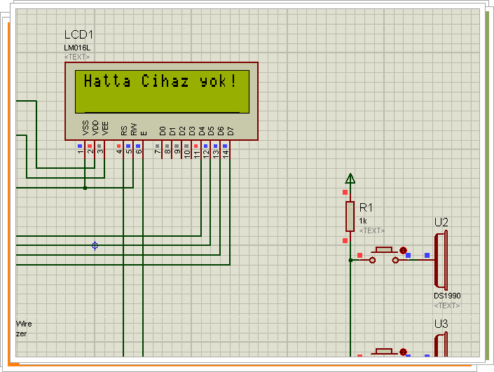
}

\_\_delay\_cycles(500000);

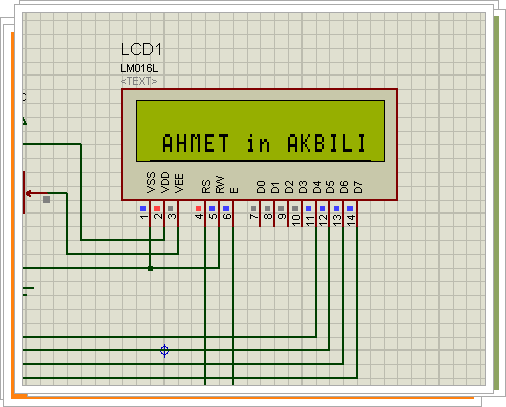
}

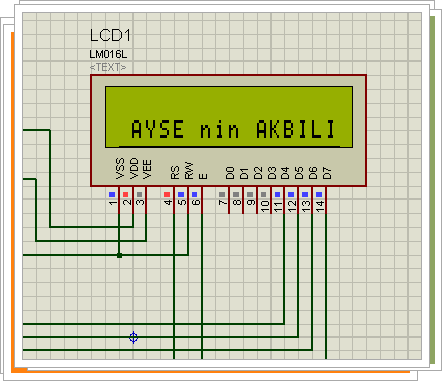
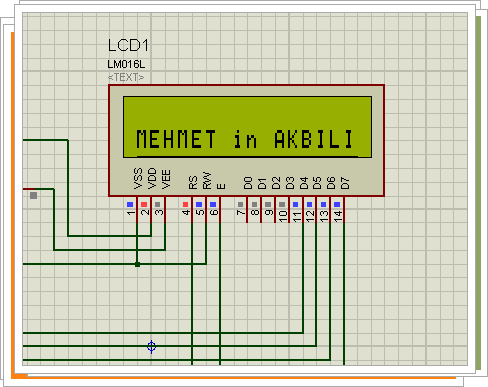
}

Uygulamamızda bir adet buzzer bulunmaktadır. Onuda eşleşen akbil olduğunda ufak bir ses üretmek için koydum. Yazılımda ise o kısımlar pasif durumdadır. İsterseniz, siz açıp deneyebilirsiniz.Yazılımdan bahsedecek olursak, ilk olarak standart ayarlar yapılmış ve lcd ekrana “DS1990a  iBUTTON” string i yazılmış ve  yaklaşık 1 sn beklenerek sonsuz döngüye girilmiştir. Sonsuz döngüde ise hatta cihaz olup olmadığı kontrol edilmiştir. Eğer cihaz bulunmuyorsa ekranda uyarı mesajı gösterilmiştir.



Eğer hattaki cihazlardan biri dokundurulursa, dokundurulan cihazın seri no su okunarak hangi kullanıcı ile eşleştiği kontrol edilerek lcd ekranda gösterilmiştir. Ardından da yaklaşık 500 msn beklenerek sonsuz döngünün başına gidilmektedir. Kullanıcılara ait cihazlar(akbiller) dokundurulduğunda lcd ekrandaki görüntüleride verecek olursak,





Görüldüğü gibi kullanıcılara ait cihazların seri numaraları okunup doğru bir şekilde kime ait olduğu tespit edilmiştir. Bu uygulamayı sizlerde kendinize göre değiştirip farklı şekillerde kullanabilirsiniz. Örneğin seri numarası tutan cihaz dokundurulduğunda, bir röle yi çektirip istediğiniz cihazı çalıştırabilirsiniz. Veya bir kapıda, sadece belirlenen kullanıcıların akbilleri ile giriş yapabilmesini sağlayabilirsiniz. Ben genel olarak OneWire haberleşme protokolünden ve DS1990a iButton cihazından bahsettim. Bir sonraki yazımda yine bir OneWire entegresi olan DS18B20 ile bir termometre uygulaması yapacağız. Uygulama hazır fakat bayram nedeniyle gecikme olabilir.

Bu arada tüm arkadaşların mübarek ramazan bayramını tebrik eder, büyüklerimin ellerinden, küçüklerimin gözlerinden öperim :) Bir sonraki dersimizde görüşmek üzere. Şimdilik Hoşçakalın…

**Ferudun GÖKCEGÖZ**

**fgokcegoz@yahoo.com**