

Fırat Üniversitesi
Mühendislik Fakültesi
Elektrik-Elektronik Mühendisliği Bölümü
Bitirme Projesi

Hazırlayanlar:
Teoman Özgür YAPAR
Ali TÜMER

Öğretim Üyesi:
Yrd.Doç.Dr.Hasan H. Balık

İçindekiler

1.Giriş

1.1Konu

1.2SCADA nedir?

2.Sıcaklık sensörü

3.Arabirim

3.1. PIC 16F877

3.2. PIC 16F877 usart

3.3. PIC 16F877 adc

4. Haberleşme Standartı Seçimi

4.1 Asenkron Seri Haberleşme

5. Sistemin Çalışması

5.1. Programlar

5.1.1. PIC 16F877 Programı

5.1.2. Visual Basic Programı

6. Kaynakça

1.Giriş

Üretim teknolojileri büyüyüp geliştikçe sistemlerin kontrolü ve gözlenebilirliği daha da önem kazanmaktadır. Üretim sürecindeki her bir sistemin anında kontrol edilmesi ve istenilen performans kriterlerine uygun üretim yapılması ve bunun eş zamanlı gerçekleşmesi problemleri, elektrik, elektronik, elektromekanik, hidrolik vb sistemler ile bilgisayar teknolojisinin uygun arabirimler vasıtası ile bir araya getirilmesiyle çözülmektedir. Öyle ki; büyük ölçekli bir üretim faaliyeti bile bir bilgisayarın çevresel bir aygıtı gibi kontrol edilebilmektedir. Elbette böyle bir tümleşik yapı, birden fazla mühendislik dalının ortak çalışmasını gerektirir. (bilgisayar, haberleşme, kontrol mühendisliği gibi.)

Sanayi devriminde gelişen makineleşme ürün standartlarını ve miktarını artırmıştır. Ancak kompleks üretim birimlerindeki verim artışı halen devam etmektedir. Bu noktada karşımıza doğru verinin doğru yerde bulunması ihtiyacı çıkmaktadır. Büyük bir üretim bandındaki olası durumların, hataların ve tüm sürecin tek bir arabirim sayesinde tek bir noktada görülebilmesinin, daha doğru karar verebilme, sonuca daha kısa zamanda ulaşabilme gibi getirileri olacaktır. Bu noktada devreye SCADA sistemler girmekte ve iş yükünün büyük bir kısmını hantal iş gücünden alıp elektronik omuzlara bırakmaktadır. *Supervisory Control and Data Acquisition* anlamına gelen scada üretim sistemlerindeki verimlilik artışı ve modernizasyonun bu gün itibarı ile geldiği son noktadır.

1.1 Konu.....: Scada ile sıcaklık kontrolü

1.2 SCADA nedir?

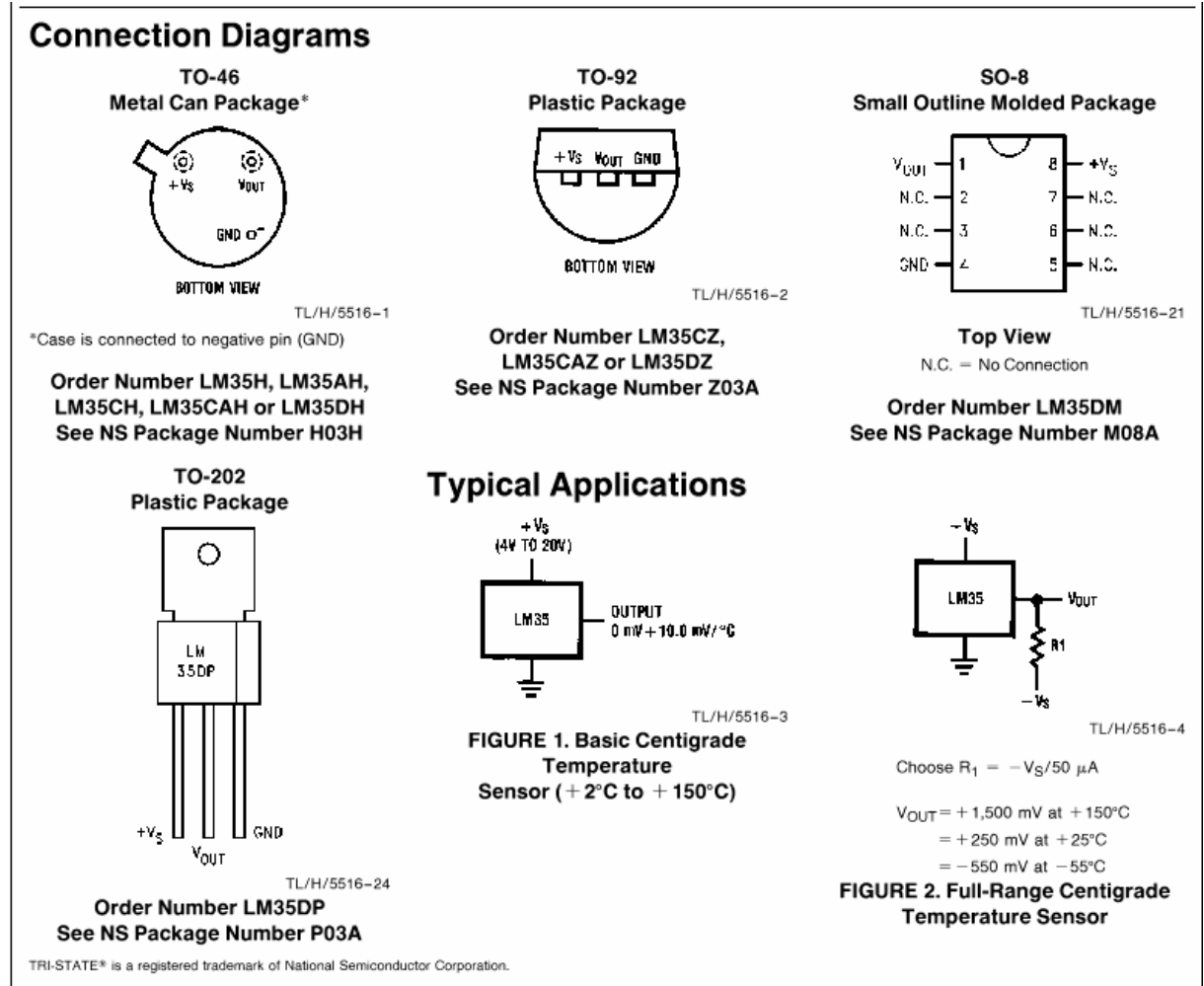
SCADA omurgalı bir sistem, işlemi, algılayıcılar (akım, gerilim, adet, sıcaklık, seviye, basınç, hız, tork vb) vasıtası ile görür. Bu değerler belli formatlarda bilgisayara aktarılır. İşin buraya kadarki kısmı donanımla ilgilidir. Bundan sonra yazılımın devreye girmesiyle kullanıcıya bu değerlerden elde edilen görsel bir ortam sunulur. Bu data akışı çift yönlüdür. Öyle ki; örneğin kullanıcının bir mouse hareketi ile rahatlıkla devasa büyüklükteki bir türbinin kapakçıklarına yön verilebilir. Bu işlemden sonra akan suyun debisi, basıncı yine aynı ekrandan izlenebilir. Sistem belli hata kodlarını sürekli tarayarak acil durum uygulamaları yapabilir. Özellikle insan hayatı açısından olumsuz şartlara sahip ortamlarda (soda, gübre fabrikaları gibi) güvenliği sağlar ve bu sayede tek noktadan kontrolün avantajlarını bize sunar. SCADA sistemi operatörler için ileri düzeyde kontrol ve gözetleme özellikleri sağlamaktadır. Genel olarak SCADA sistemi, uygulamada şu imkanları sağlayabilir.

- Kullanıcı tarafından tanımlanmış işletmeye ait parametreler (seviye, sıcaklık, basınç, dijital sinyaller, vana ve motor durumları, sistem durumu vb.) vasıtasıyla işletmenin takibi
- Reçete ekranları vasıtasıyla, üretim reçetelerinin girilmesi ve işleyen reçeteler hakkında operatörün bilgilendirilmesi
- Parametre ekranları vasıtasıyla, sistem için gerekli olan limit değerlerin (set-point, alt ve üst alarm değerleri) girilmesi
- P, I, D parametrelerinin girilebilmesi ve gözetlenmesi
- İşletme değerlerinin tarihsel ve gerçek zamanlı trendlerinin tutulması
- Anlık ve periyodik raporların (üretim, reçete, stok vb.) alınması
- Otomatik çalışan sisteme, SCADA ekranlarından manuel müdahale imkanı
- Alarm ve durumların gösterilmesi ve yazıcıya ve/veya veri tabanına kayıt edilmesi
- İleri düzeyde kalite kontrol (örneğin istatistiksel proses kontrol) desteği

2.SICAKLIK SENSÖRÜ

Sistemde LM 35 entegresi kullanıldı.

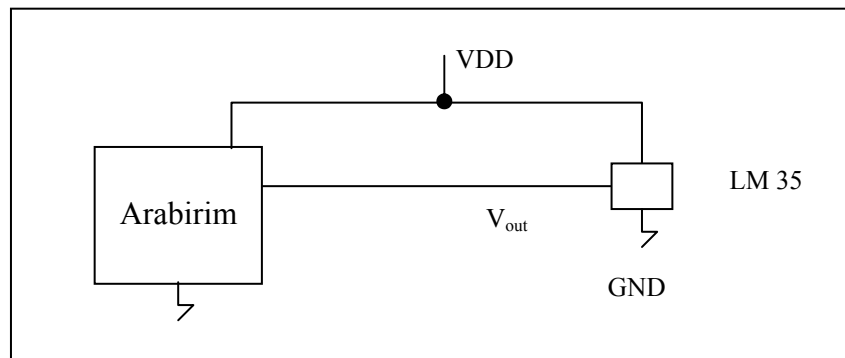
-50 ile 150 derece arasındaki sıcaklıklarda değişen her dereceye karşılık 10 mV'luk çıkış göstermekte. Herhangi bir kalibrasyona ihtiyaç duymaması (0 derece= 0 Volt), 4-20 V geniş besleme aralığına sahip olması ve basit bağlantı özellikleri LM 35'i bu sistem için en uygun sıcaklık sensörü haline getiriyor. Tablo 1'de LM35 bağlantı diyagramları ,Tablo 2'de LM35in elektriksel karakteristikleri ve şekil 1'de ise LM35 in arabirim bağlantısı görülmektedir



Tablo 1: LM35 bağlantı diyagramı

| Electrical Characteristics (Note 1) (Note 6) (Continued) | | | | | | | | |
|--|---|--------------|-----------------------|-----------------------|--------------|-----------------------|-----------------------|------------------------------|
| Parameter | Conditions | LM35 | | | LM35C, LM35D | | | Units (Max.) |
| | | Typical | Tested Limit (Note 4) | Design Limit (Note 5) | Typical | Tested Limit (Note 4) | Design Limit (Note 5) | |
| Accuracy, LM35, LM35C (Note 7) | $T_A = +25^\circ\text{C}$ | ± 0.4 | ± 1.0 | | ± 0.4 | ± 1.0 | | $^\circ\text{C}$ |
| | $T_A = -10^\circ\text{C}$ | ± 0.5 | | | ± 0.5 | | ± 1.5 | $^\circ\text{C}$ |
| | $T_A = T_{\text{MAX}}$ | ± 0.8 | ± 1.5 | | ± 0.8 | | ± 1.5 | $^\circ\text{C}$ |
| | $T_A = T_{\text{MIN}}$ | ± 0.8 | | ± 1.5 | ± 0.8 | | ± 2.0 | $^\circ\text{C}$ |
| Accuracy, LM35D (Note 7) | $T_A = +25^\circ\text{C}$ | | | | ± 0.6 | ± 1.5 | | $^\circ\text{C}$ |
| | $T_A = T_{\text{MAX}}$ | | | | ± 0.9 | | ± 2.0 | $^\circ\text{C}$ |
| | $T_A = T_{\text{MIN}}$ | | | | ± 0.9 | | ± 2.0 | $^\circ\text{C}$ |
| Nonlinearity (Note 8) | $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$ | ± 0.3 | | ± 0.5 | ± 0.2 | | ± 0.5 | $^\circ\text{C}$ |
| Sensor Gain (Average Slope) | $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$ | $+ 10.0$ | $+ 9.8,$ $+ 10.2$ | | $+ 10.0$ | | $+ 9.8,$ $+ 10.2$ | mV/ $^\circ\text{C}$ |
| Load Regulation (Note 3) $0 \leq I_L \leq 1 \text{ mA}$ | $T_A = +25^\circ\text{C}$ | ± 0.4 | ± 2.0 | | ± 0.4 | ± 2.0 | | mV/mA |
| | $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$ | ± 0.5 | | ± 5.0 | ± 0.5 | | ± 5.0 | mV/mA |
| Line Regulation (Note 3) | $T_A = +25^\circ\text{C}$ | ± 0.01 | ± 0.1 | | ± 0.01 | ± 0.1 | | mV/V |
| | $4\text{V} \leq V_S \leq 30\text{V}$ | ± 0.02 | | ± 0.2 | ± 0.02 | | ± 0.2 | mV/V |
| Quiescent Current (Note 9) | $V_S = +5\text{V}, +25^\circ\text{C}$ | 56 | 80 | | 56 | 80 | | μA |
| | $V_S = +5\text{V}$ | 105 | | 158 | 91 | | 138 | μA |
| | $V_S = +30\text{V}, +25^\circ\text{C}$ | 56.2 | 82 | | 56.2 | 82 | | μA |
| | $V_S = +30\text{V}$ | 105.5 | | 161 | 91.5 | | 141 | μA |
| Change of Quiescent Current (Note 3) | $4\text{V} \leq V_S \leq 30\text{V}, +25^\circ\text{C}$ | 0.2 | 2.0 | | 0.2 | 2.0 | | μA |
| | $4\text{V} \leq V_S \leq 30\text{V}$ | 0.5 | | 3.0 | 0.5 | | 3.0 | μA |
| Temperature Coefficient of Quiescent Current | | $+ 0.39$ | | $+ 0.7$ | $+ 0.39$ | | $+ 0.7$ | $\mu\text{A}/^\circ\text{C}$ |
| Minimum Temperature for Rated Accuracy | In circuit of Figure 1, $I_L = 0$ | $+ 1.5$ | | $+ 2.0$ | $+ 1.5$ | | $+ 2.0$ | $^\circ\text{C}$ |
| Long Term Stability | $T_J = T_{\text{MAX}}$, for 1000 hours | ± 0.08 | | | ± 0.08 | | | $^\circ\text{C}$ |

Tablo 2: LM35 elektriksel karakteristikleri



Şekil 1: LM 35 arabirim bağlantısı

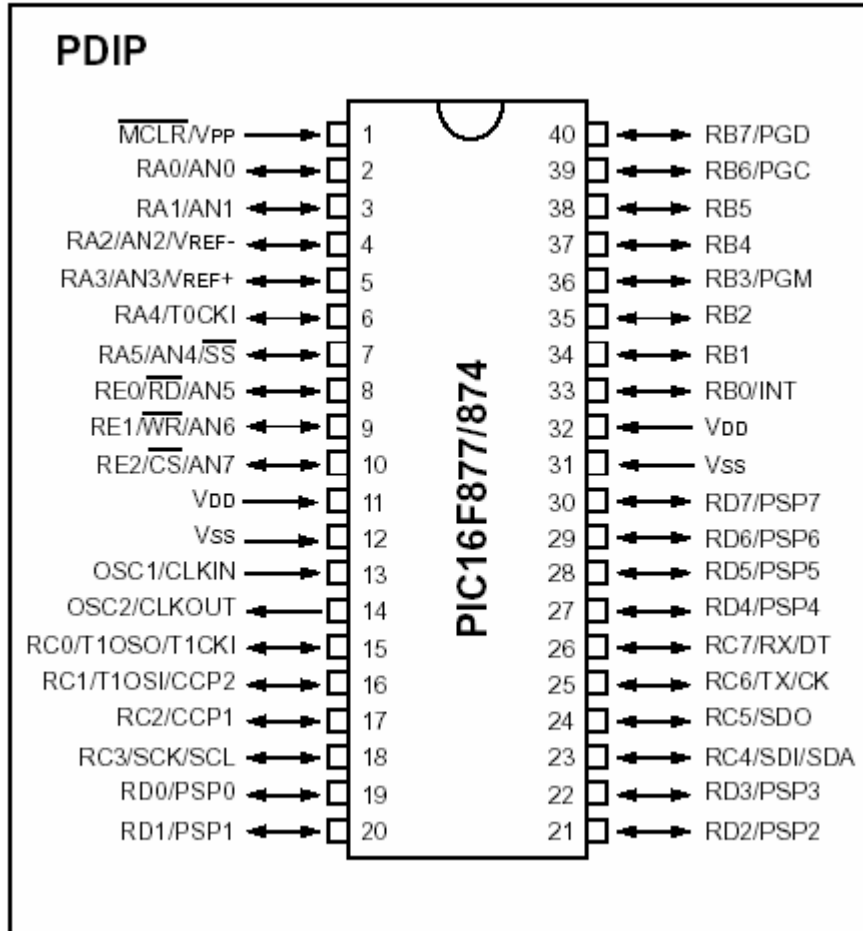
Sıcaklık değerleri sonuçta bilgisayara aktarılacağından bir analog dijital çeviriciye ihtiyacımız olacak. Bunu da arabirim içerisinde gerçekleştirebiliriz. Buradaki arabirim kullandığımız pic 16f877 olacaktır. Bilgisayara aktarılan bilgi eşdeğer dijital sinyallerden ibaret olacaktır.

3. ARABİRİM

Sistemdeki işin büyük bir kısmı arabirim tarafından gerçekleştirilmektedir. Sıcaklık bilgisi için gereken analog dijital çevirci, sürme uçları ve PC haberleşmesi için gereken donanım arabirim devresi içerisinde yer almaktadır. Bunlara ek olarak, içerisinde haberleşme trafiğinin sürekliliğini sağlamak için küçük çapta hafıza modülleri de bulunmalıdır. Bahsedilen özelliklere sahip bir devre tasarlamak hem analog hem de sayısal açıdan mümkündür ve sayısız kombinasyona sahiptir. Mikroişlemci omurgalı bir kart aklıyla ilk gelen model olmakla birlikte, sayılan diğer bileşenleri de tasarlamak gerekir. (ADC, PC haberleşme arabirimi vb.) Bütün bunları bünyesinde barındıran kapalı devre programlanabilir entegreler de mevcuttur. En bilineni Microchip firmasının ürettiği PIC mikroişlemcileridir. Çok çeşitli konfigürasyonlarda üretilen Pic mikroişlemcileri sahip olduğu özellikler sayesinde TTL seviyesindeki hemen hemen tüm elektronik uygulama için aklıyla gelen ilk yapı olmaktadır. Klasik mikroişlemci kartlarındaki tüm donanımı ve fazlasını bünyesinde bulundurmasının yanında düşük güç tüketmesi özelliği ile de özellikle küçük boyutlu ve taşınabilir devreler için en uygun model olmaktadır. Biz de sistemdeki arabirim devresini PIC16F877 işlemcisi üzerine kurduk.

3.1. PIC16F877 Mikroişlemcisi

40 pin, 8 bit, flash tip mikro denetleyicidir. Bünyesinde USART, 8 adet 10 bit ADC, 3 adet birbirinden bağımsız zamanlayıcı, 14 farklı kesme vektörü, 16 bit karşılaştırıcı, 16 bit tutucu, 10 bit PWM, 8 bit harici paralel portu barındırır. Her birine farklı görevler atanabilen 33 giriş-çıkış uç içeren denetleyici toplam 40 pindir. Şekil 2’de pin diyagramı görülmektedir.



Şekil 2: PIC 16f877 pin diyagramı

Kullanılacak özelliklerden usart ve adc'yi açıklamaya çalışalım...

3.2. PIC16F877 Usart

PIC16F877'de seri port olarak veri göndermek (transmit) için RC6 pini, veri almak (receive) için RC7 pini kullanılmakta. Bu pinlerin statülerini ve tüm seri haberleşmeyi kontrol eden 2 yazmaç var. Seri haberleşme yapmak için öncelikle bu pinlerin RX ve TX olarak ayarlanması gerekiyor. Sonrasında yine haberleşme ile ilgili bilgilerin (baud hızı, stop bit sayısı, parity durumu) bu 2 yazmaca yazılması gerekiyor.

REGISTER 10-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS 98h)

| | | | | | | | |
|-------|-------|-------|-------|-----|-------|------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R-1 | R/W-0 |
| CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D |
| bit 7 | | | | | | | bit 0 |

Bit 7: CSRC :Zamanlamanın kaynağını belirliyor. Asenkron haberleşmede değeri önemli değil (0-1)

Bit 6: TX9 :9 bit veri gönderme izin biti.

1 : 9 bit gönderim seçer

0 : 8 bit gönderim seçer

Bit 5: TXEN :Transmit izin biti.

1 : gönder

0 : gönderme

Bit 4: SYNC :Usart mod seçme biti

1 : senkron mod

0 : asenkron mod

Bit 3: - : Kullanılmıyor. Okunduğunda sıfır değeri veriyor.

Bit 2: BRGH :Hızlı-yavaş baud hızı seçme biti

1 : yüksek hız

0 : düşük hız

Bit 1: TRMT :Transmit kaydırma yazmacı durum biti (shift register)

1 : TRM boş

0 : TRM dolu

Bit 0 : TX9D : 9 bit haberleşmenin 9. biti. Parity biti de olabilir.

REGISTER 10-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)

| | | | | | | | |
|-------|-------|-------|-------|-------|------|------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | R-0 | R-x |
| SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D |
| bit 7 | | | | | | | bit 0 |

Bit 7: SPEN :Seri port seçme biti

1 : RC7 ve RC6 pinlerini seri port için ayarlar

0 : Seri port ayarlaması yok

Bit 6: RX9 :9 bit veri alma izin biti.

1 : 9 bit alma seçer

0 : 8 bit alma seçer

Bit 5: SREN :Tek data alma izin biti. Asenkron moda kullanılmıyor

Bit 4: CREN :Devamlı alma izin biti

1 : devamlı alma aktif

0 : devamlı alma pasif

Bit 3: ADDEN :RX9 biti 1 iken adres sorgusu yapmak için kullanılıyor. Bu sistemde kullanılmayacak.

Bit 2: FERR :Veri kayma hata biti (framing error)

1 : kayma hatası var

0 : kayma hatası yok

Bit 1: OERR :Taşma hata biti (overrun)

1 : taşma var

0 : taşma yok

Bit 0 : RX9D : 9 bit haberleşmenin 9. biti. Parity biti de olabilir.

İşlemci yazılımının başında bu registerlere kullanılacak konfigürasyona göre gereken değerler yazılacaktır. Baud hızı için ise SPBRG registeri kullanılıyor. Buraya yazılan 8 bitlik değer ile baud hızı *tablo 3*'teki gibi yapılıyor.

| SYNC | BRGH = 0 (Low Speed) | BRGH = 1 (High Speed) |
|------|--|---------------------------------|
| 0 | (Asynchronous) Baud Rate = $F_{OSC}/(64(X+1))$ | Baud Rate = $F_{OSC}/(16(X+1))$ |
| 1 | (Synchronous) Baud Rate = $F_{OSC}/(4(X+1))$ | N/A |

Tablo 3: baud hızı formülasyonu

Burada F_{OSC} kullanılan osilatör frekansını, X ise SPBRG registerine yazılması gereken değeri temsil ediyor. Elbetteki maksimum 20 MHz saat frekansında çalışan bir denetleyicinin bütün baud hızlarında, X değerlerinin tam doğru değerler vermesi mümkün değildir. *Tablo 4* ve *tablo 5*'de BRGH'ın 0 ve 1 olduğu durumlar için, çeşitli saat frekanslarında ve baud hızlarındaki bağıl hata görülmektedir.

| BAUD RATE (K) | Fosc = 20 MHz | | | Fosc = 16 MHz | | | Fosc = 10 MHz | | |
|---------------|---------------|---------|-----------------------|---------------|---------|-----------------------|---------------|---------|-----------------------|
| | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) |
| 0.3 | - | - | - | - | - | - | - | - | - |
| 1.2 | 1.221 | 1.75 | 255 | 1.202 | 0.17 | 207 | 1.202 | 0.17 | 129 |
| 2.4 | 2.404 | 0.17 | 129 | 2.404 | 0.17 | 103 | 2.404 | 0.17 | 64 |
| 9.6 | 9.766 | 1.73 | 31 | 9.615 | 0.16 | 25 | 9.766 | 1.73 | 15 |
| 19.2 | 19.531 | 1.72 | 15 | 19.231 | 0.16 | 12 | 19.531 | 1.72 | 7 |
| 28.8 | 31.250 | 8.51 | 9 | 27.778 | 3.55 | 8 | 31.250 | 8.51 | 4 |
| 33.6 | 34.722 | 3.34 | 8 | 35.714 | 6.29 | 6 | 31.250 | 6.99 | 4 |
| 57.6 | 62.500 | 8.51 | 4 | 62.500 | 8.51 | 3 | 52.083 | 9.58 | 2 |
| HIGH | 1.221 | - | 255 | 0.977 | - | 255 | 0.610 | - | 255 |
| LOW | 312.500 | - | 0 | 250.000 | - | 0 | 156.250 | - | 0 |

Tablo:4

| BAUD RATE (K) | Fosc = 4 MHz | | | Fosc = 3.6864 MHz | | |
|---------------|--------------|---------|-----------------------|-------------------|---------|-----------------------|
| | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) |
| 0.3 | 0.300 | 0 | 207 | 0.3 | 0 | 191 |
| 1.2 | 1.202 | 0.17 | 51 | 1.2 | 0 | 47 |
| 2.4 | 2.404 | 0.17 | 25 | 2.4 | 0 | 23 |
| 9.6 | 8.929 | 6.99 | 6 | 9.6 | 0 | 5 |
| 19.2 | 20.833 | 8.51 | 2 | 19.2 | 0 | 2 |
| 28.8 | 31.250 | 8.51 | 1 | 28.8 | 0 | 1 |
| 33.6 | - | - | - | - | - | - |
| 57.6 | 62.500 | 8.51 | 0 | 57.6 | 0 | 0 |
| HIGH | 0.244 | - | 255 | 0.225 | - | 255 |
| LOW | 62.500 | - | 0 | 57.6 | - | 0 |

| BAUD RATE (K) | Fosc = 20 MHz | | | Fosc = 16 MHz | | | Fosc = 10 MHz | | |
|---------------|---------------|---------|-----------------------|---------------|---------|-----------------------|---------------|---------|-----------------------|
| | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) |
| 0.3 | - | - | - | - | - | - | - | - | - |
| 1.2 | - | - | - | - | - | - | - | - | - |
| 2.4 | - | - | - | - | - | - | 2.441 | 1.71 | 255 |
| 9.6 | 9.615 | 0.16 | 129 | 9.615 | 0.16 | 103 | 9.615 | 0.16 | 64 |
| 19.2 | 19.231 | 0.16 | 64 | 19.231 | 0.16 | 51 | 19.531 | 1.72 | 31 |
| 28.8 | 29.070 | 0.94 | 42 | 29.412 | 2.13 | 33 | 28.409 | 1.36 | 21 |
| 33.6 | 33.784 | 0.55 | 36 | 33.333 | 0.79 | 29 | 32.895 | 2.10 | 18 |
| 57.6 | 59.524 | 3.34 | 20 | 58.824 | 2.13 | 16 | 56.818 | 1.36 | 10 |
| HIGH | 4.883 | - | 255 | 3.906 | - | 255 | 2.441 | - | 255 |
| LOW | 1250.000 | - | 0 | 1000.000 | - | 0 | 625.000 | - | 0 |

Tablo:5

| BAUD RATE (K) | Fosc = 4 MHz | | | Fosc = 3.6864 MHz | | |
|---------------|--------------|---------|-----------------------|-------------------|---------|-----------------------|
| | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) |
| 0.3 | - | - | - | - | - | - |
| 1.2 | 1.202 | 0.17 | 207 | 1.2 | 0 | 191 |
| 2.4 | 2.404 | 0.17 | 103 | 2.4 | 0 | 95 |
| 9.6 | 9.615 | 0.16 | 25 | 9.6 | 0 | 23 |
| 19.2 | 19.231 | 0.16 | 12 | 19.2 | 0 | 11 |
| 28.8 | 27.798 | 3.55 | 8 | 28.8 | 0 | 7 |
| 33.6 | 35.714 | 6.29 | 6 | 32.9 | 2.04 | 6 |
| 57.6 | 62.500 | 8.51 | 3 | 57.6 | 0 | 3 |
| HIGH | 0.977 | - | 255 | 0.9 | - | 255 |
| LOW | 250.000 | - | 0 | 230.4 | - | 0 |

Burada bizim ilgilendiğimiz frekans 4 MHz dir. Baud hızını ise 9600 seçiyoruz. Bu şekilde teknik olarak bağıl haberleşme hatasını sıfır kabul ediyoruz..

3.3. PIC16F877 ADC

Analog dijital çevirici modülü kullanmak için ADCON0 ve ADCON1 registerlerini işleme dahil etmemiz gerekiyor. İlgili ayarlar bu registerlerden yapılıyor. 10 bit için 8 bitlik 2 register kullanılıyor. ADRESH ve ADRESL registerleri çevrim değerini tutuyor.

REGISTER 11-1: ADCON0 REGISTER (ADDRESS: 1Fh)

| | | | | | | | |
|-------|-------|-------|-------|-------|---------|-----|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 |
| ADCS1 | ADCS0 | CHS2 | CHS1 | CHS0 | GO/DONE | — | ADON |
| bit 7 | | | | | | | bit 0 |

Bit 7-6: ADCS1, ADCS2 :ADC saati seçme biti

00 : $F_{OSC}/2$

01 : $F_{OSC}/8$

10 : $F_{OSC}/32$

11 : F_{RC} dahili RC osilatör

Bit 5-3: CHS2,CHS1,CHS0 :Analog kanal seçme biti

000 : kanal 0, (RA0)

001 : kanal 1, (RA1)

010 : kanal 2, (RA2)

011 : kanal 3, (RA3)

100 : kanal 4, (RA5)

101 : kanal 5, (RE0)

110 : kanal 6, (RE1)

111 : kanal 7, (RE2)

Bit 2: $\overline{GO/DONE}$:A/D çevrim durum biti

1 : çevrimi başlatmak için 1 yapılır, çevrim tamamlanınca otomatik olarak 0 olur.

0 : ADC meşgul değil

Bit 1: - :Kullanılmıyor. 0 okunur.

Bit 0: ADON :A/D modül açık biti

1 : modül açık, işlem sürüyor

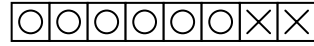
0 : modül kapalı, çevrim akımı yok

REGISTER 11-2: ADCON1 REGISTER (ADDRESS 9Fh)

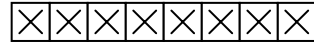
| | | | | | | | |
|-------|-----|-------|-----|-------|-------|-------|-------|
| U-0 | U-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| ADFM | — | — | — | PCFG3 | PCFG2 | PCFG1 | PCFG0 |
| bit 7 | | | | | | | bit 0 |

Bit 7: ADFM :A/D sonuç format biti

1 : sağa dayalı sonuç formatı, ADRESH'in en önemli 6 biti 0 okunur



ADRESH



ADRESL

0 : sola dayalı sonuç formatı, ADRESL'nin en önemsiz 6 biti 0 okunur



ADRESH



ADRESL

Bit 6-4: - :Kullanılmıyor. 0 okunuyor.

Bit 3-0: PCFG3, PCFG2, PCFG1, PCFG0 : A/D port konfigürasyon kontrol biti

| PCFG3: PCFG0 | AN7 ⁽¹⁾ RE2 | AN6 ⁽¹⁾ RE1 | AN5 ⁽¹⁾ RE0 | AN4 RA5 | AN3 RA3 | AN2 RA2 | AN1 RA1 | AN0 RA0 | VREF+ | VREF- | CHAN/ Refs ⁽²⁾ |
|-----------------|---------------------------|---------------------------|---------------------------|------------|------------|------------|------------|------------|-------|-------|------------------------------|
| 0000 | A | A | A | A | A | A | A | A | VDD | Vss | 8/0 |
| 0001 | A | A | A | A | VREF+ | A | A | A | RA3 | Vss | 7/1 |
| 0010 | D | D | D | A | A | A | A | A | VDD | Vss | 5/0 |
| 0011 | D | D | D | A | VREF+ | A | A | A | RA3 | Vss | 4/1 |
| 0100 | D | D | D | D | A | D | A | A | VDD | Vss | 3/0 |
| 0101 | D | D | D | D | VREF+ | D | A | A | RA3 | Vss | 2/1 |
| 011x | D | D | D | D | D | D | D | D | VDD | Vss | 0/0 |
| 1000 | A | A | A | A | VREF+ | VREF- | A | A | RA3 | RA2 | 6/2 |
| 1001 | D | D | A | A | A | A | A | A | VDD | Vss | 6/0 |
| 1010 | D | D | A | A | VREF+ | A | A | A | RA3 | Vss | 5/1 |
| 1011 | D | D | A | A | VREF+ | VREF- | A | A | RA3 | RA2 | 4/2 |
| 1100 | D | D | D | A | VREF+ | VREF- | A | A | RA3 | RA2 | 3/2 |
| 1101 | D | D | D | D | VREF+ | VREF- | A | A | RA3 | RA2 | 2/2 |
| 1110 | D | D | D | D | D | D | D | A | VDD | Vss | 1/0 |
| 1111 | D | D | D | D | VREF+ | VREF- | D | A | RA3 | RA2 | 1/2 |

Bu sistemde sadece 1 tane A/D kanal gerektiğinden 1110 seçmemiz yeterli olacaktır..

4. HABERLEŞME STANDARTI SEÇİMİ

Bilgisayarlar temel olarak paralel ve seri olmak üzere iki tür haberleşme standardını kullanır. Her ikisinin de birbirine göre üstün yanları vardır. Paralel veri iletimi 8 bitlik veri yolu kullanır. Bu da her bir zaman çevriminde (clock puls) 256 (2^8) karakterden bir tanesinin gönderilmesini olanaklı kılar. Seri veri iletiminin veri yolu tek bitlidir. Bu tek bit, zaman üzerinde paylaşılarak her bir zaman çevriminde bir karakterin yalnız bir bitinin gönderilmesini olanaklı kılar. Bu paralel haberleşmeyi üstün kılıyor gibi gözükmesine rağmen, daha az veri yolu kullanmak bir mühendislik düşüncesi gereğidir. Her iki haberleşme standardını kıyaslamak için tablo 6'ya göz atalım:

| <i>Paralel iletim</i> | <i>Seri iletim</i> |
|---|--|
| Çok hızlı bilgi gönderilip – alınabilir | Paralel kadar hızlı değildir |
| Kullanılan kablo uzunluğu 9 feet olabilir. | Kullanılan kablo uzunluğu kısıtlanmaz. |
| Paralel iletim kabloları çokludur. | Seri iletim kabloları genelde çifttir. |
| Bilginin bir karakteri bir kerede gönderebilir. | Karakterin bir biti bir kerede gönderilir. |
| Senkronizasyon için bir hat işgal eder | Asenkron çalışmayı destekler |

Tablo 6: Paralel ve seri haberleşme standartları karşılaştırması

Endüstrideki bir çok uygulama için seri haberleşme yeterli, çoğu zaman da gereklidir. Bu uygulama için de seri haberleşme hızı yeterli olmaktadır. Bu yüzden seri haberleşme standardını seçtik.

4.1. Asenkron Seri Haberleşme Standardı

Bir karakterin bitleri tek bir hat üzerinden sırayla gönderilir. Karakterin başında gönderilen başlama bitinden sonra alıcı ve verici birbirinden bağımsızdır. Verici, bitleri belirlenmiş bir hızda gönderirken, alıcı da teorik olarak aynı hızda bunları okur. Asenkron seri haberleşmenin önemli terimlerini şu şekilde sıralayabiliriz:

Baud Rate: Veri iletim hızıdır. 1 saniyede iletilen bit adedine denir. Standart olarak veri hızları 300,600,1200,2400,4800,9600,19200, ... şeklindedir.

Start Bit: Seri haberleşmede veriler senkron yada asenkron olarak iletilebilirler. Senkron iletimde belirli bir başlangıçtan sonra veriler sıralı olarak ard arda gelirler. Uzun bir dosyanın iletiminde bu yol kullanılabilir. Asenkron iletimde ise bir veri gönderildikten sonra yeni bir veri belirli olmayan bir zamanda gelir. İşte bu yeni verinin başlangıcı start biti ile bildirilir.

Stop Bit: Gönderilen verinin bittiğini ifade eden bittir. Bu biti alan alıcı yeni bir veri için start bitini gözlemeye başlar. haberleşmenin tipine göre 1 yada 2 bit uzunluğunda olabilir.

Eşlik Biti (Parity Bit): Bir çeşit hata denetim bitidir.start ve 8 bitlik bir veri iletildikten sonra stop biti gönderilmeden parity gönderilir. iletilen 8 bitlik veride 1 lerin sayısı çift ise EVEN tek ise ODD biti gönderilir. Alıcı bu biti kontrol ederek alınan verinin doğru olup olmadığını kontrol eder. 1 parity 1 bit uzunluğundadır.

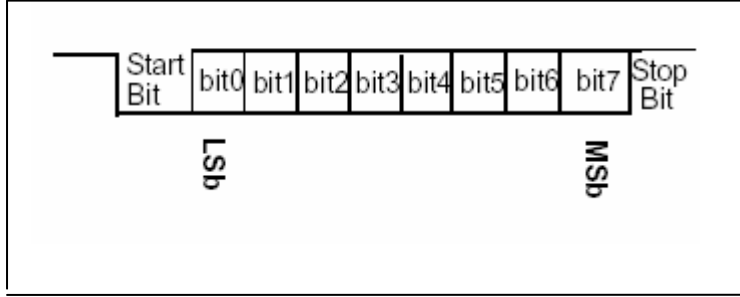
Siplex: Bilginin sadece bir yönde aktığı durumdur.

Half duplex: Bilginin farklı zamanlarda çift yönde akabildiği durumdur.

Full duplex: Bilginin aynı anda çift yönde akabildiği durumdur.

Şekil 3’de asenkron seri haberleşme zaman diyagramı görülmektedir.

Şekil:3



5. SİSTEMİN ÇALIŞMASI

Buraya kadar sistemimizde bulunan elemanlar ve arabirim hakkında bilgi verdik. Biraz sistemimizin nasıl çalıştığı hakkında bilgi vermeye çalışalım.

Sistemin PIC ile yapılan kontrol birimi default değeri OFF konumunda başlıyor. Hiçbir işlem yapılmıyor.

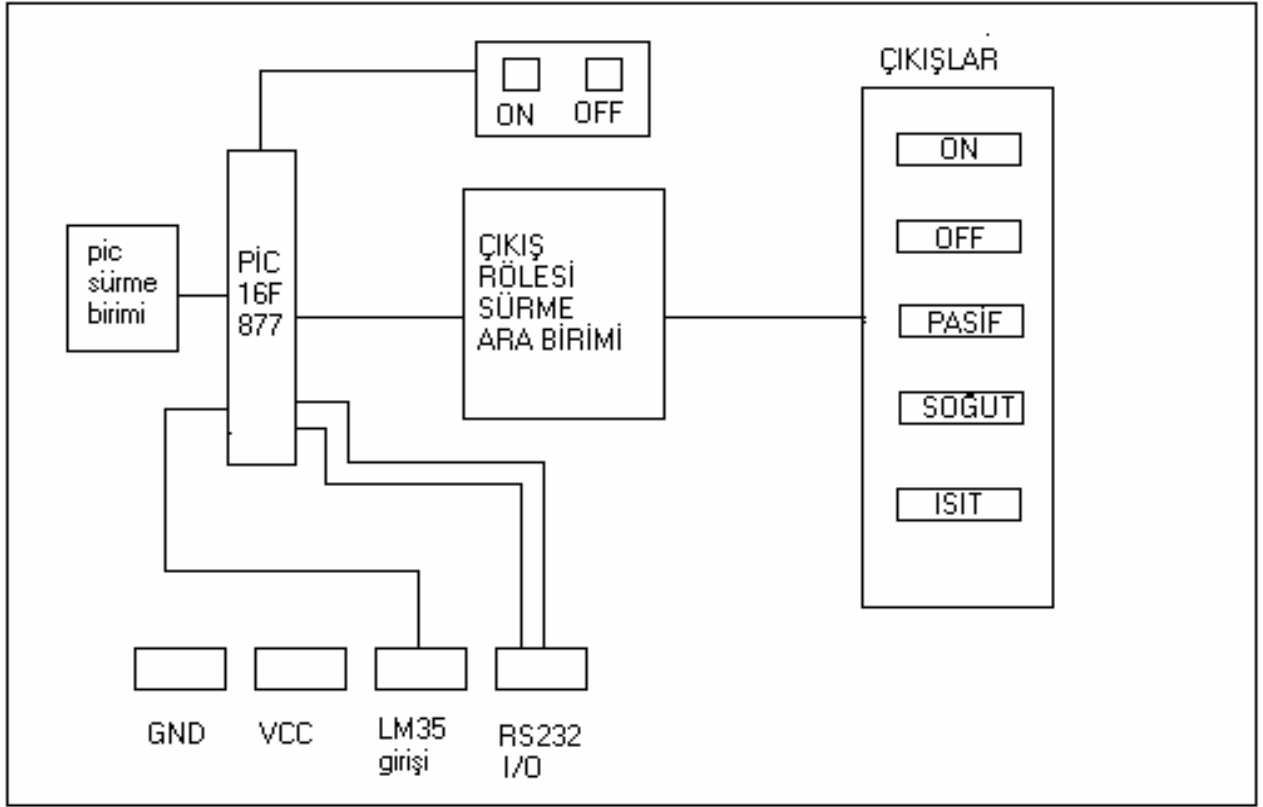
Bilgisayar arabirim ile ilgili sayfayı açtığımızda PIC’li arabirim “ON” konumuna geçiyor ve otomatik modda o anki oda sıcaklığını (ortam sıcaklığı) okuyor. İstenen sıcaklıkta bir değer yazılmadığı sürece bir işlem yapmadan bekliyor. İstenen sıcaklık değeri girilince sistem otomatik olarak ısıtma veya soğutma işlemini yapıyor.(Buradaki tek problem LM35’in çok değişken bir değer göndermesi ile ölçmede sorun yaşanması)

Anında aç kapaları önlemek için 3-4 sn’lik bir gecikme yapılıyor. Böylelikle bu dalgalanmanın kısmen önüne geçerek her an aç-kapa yapılmamış oluyor.

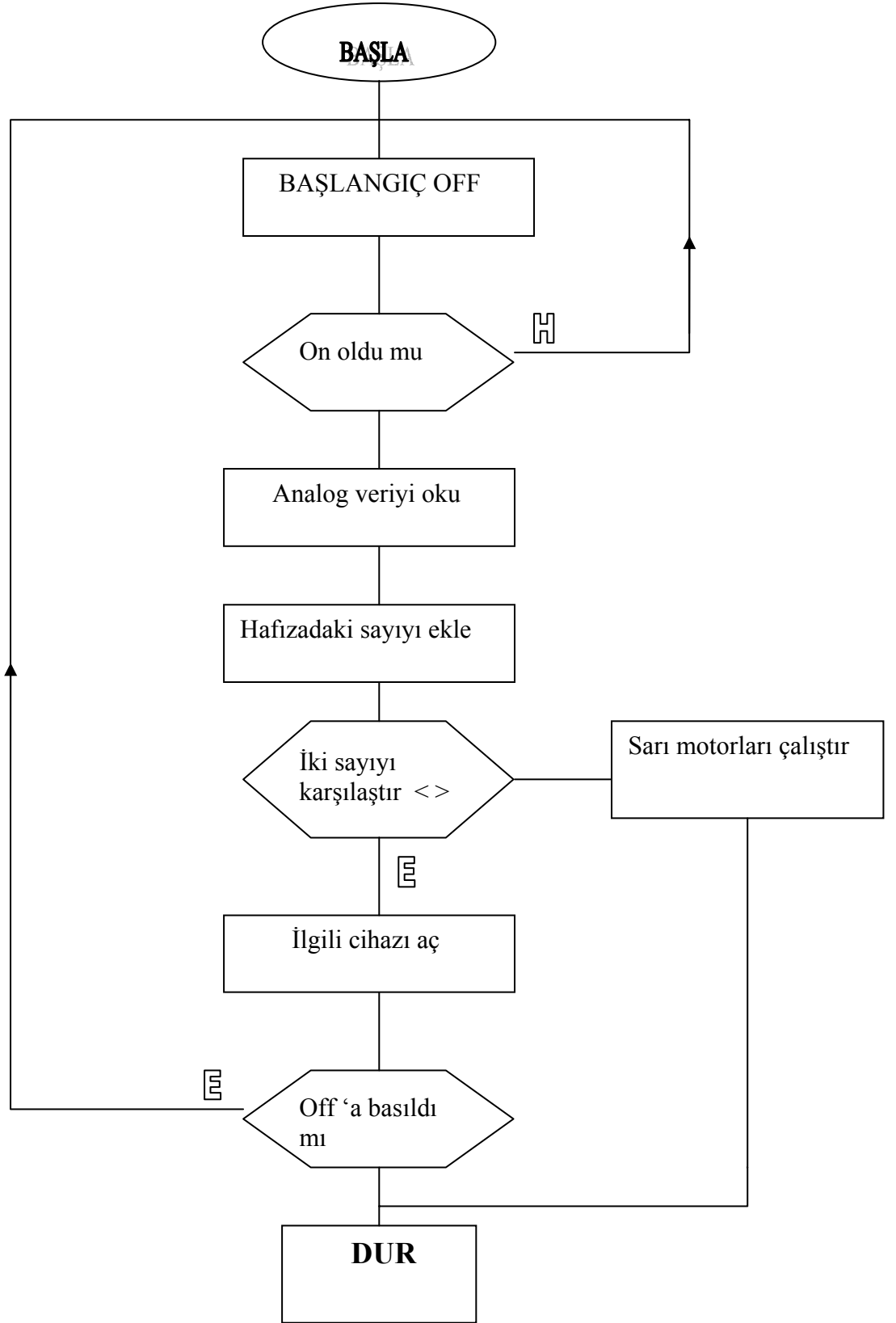
Sistem menüdeki manuel moda geçince ısıtma ve soğutma işlemleri ortam sıcaklığı ne olursa olsun tamamen kullanıcıya bırakılıyor.

OFF konumuna geçince ise tüm sistem tekrar pasif duruma geçiyor.

Aşağıda ise sistemde kullandığımız devreye ait bir blok diyagramı ve sistemin nasıl çalıştığına dair bir akış diyagramı verilmiştir.



Şekil 4: Devrenin blok diyagramı



5.1 Programlar

Sistemimizi gerçekleştirirken arabirim olarak PIC 16F877 kullandığımızı daha önce belirtmiştik. Kullanmış olduğumuz bu mikrodenetleyici programlanmadığı sürece çalışmaz. Dolayısıyla bu mikrodenetleyici istenilen giriş ve çıkışları elde etmemiz için programlanması gerekir. Yazılmış olan bu program aşağıda verilmiştir.

Devremizin nasıl çalıştığını görmek ve ona müdahale etmek SCADA'nın gerekliliği idi. Bu işlevi yerine getirmek için visual basic ile bir program yazdık. Bu programın kodları da yine aşağıda verilmiştir.

5.1.1 PIC 16F877 programı

```
INCLUDE"MODEDEFS.BAS"
```

```
' Değişkenler yerleştiriliyor.
```

```
x var word
```

```
y1 var word
```

```
y con 66
```

```
Z VAR WORD
```

```
w var word
```

```
trisb.5=0
```

```
trisd=0
```

```
ADCON1 = $81
```

```
Pause 100
```

```
Goto mainloop
```

```
' Subroutine to read a/d convertor
```

```
getad:
```

```
Pauseus 50
```

```
ADCON0.2 = 1
```

```
Pauseus 50
```

```
Return
```

```
' Subroutine to get pot x value
```

```
getx:
```

```
ADCON0 = $11
```

```
Gosub getad
```

```
z=adresl
```

```
Return
```

```
mainloop:
```

```
Gosub getx
```

```
pause 100
```

```
serout2 portb.1,16468,[" ",#z]
```

```
pause 100
```

```
serin2 portb.7,16468,[y1]
```

```
'aktif sıcaklık=İstenen sıcaklık gönderilen 66
```

```
if y=y1 then
```

```
'eşitlik ledi
```

```
portb.5=1
```

```
portd.6=0
```

```
portd.1=0
```

```
endif
```



```

'aktif sıcaklık<istenen sıcaklık ısıtılacak 65
if y>y1 then
'ısıtma ledi
portd.1=1
portb.5=0
portd.6=0
endif
'aktif sıcaklık>istenen sıcaklık soğutulacak 67
if y<y1 then
'soğutma ledi
portd.6=1
portb.5=0
portd.1=0
endif
Goto mainloop
End

```

5.1.2 Visual Basic Programı

```

Private Sub Form_Load()
Dim x As Variant
Dim dosya
If MSComm1.PortOpen = False Then 'port açık değilse aç
MSComm1.PortOpen = True 'bu kontrolü yaptırmassan ve port açıksa
End If 'hata verir.
txtissic.Text = "" 'başlangıçta istenen sıcaklık değeri boş olmalı
Ttxtaksic.Text = txtissic.Text 'bu boş değeri başlangıçta aktif sıcaklığa atar
Timer4.Enabled = True 'başlarken on konumuna geç
frmOn.BackColor = &HFF00&
frmOff.BackColor = &HFFFF&
If Option2.Value = True Then 'başlangıçta otomatik modda çalış

```

```

cmdisit.Enabled = False 'tuşların hepsi pasif durumda olacak
cmdSog.Enabled = False 'tuşların hepsi pasif durumda olacak
Cmdstop.Enabled = False 'tuşların hepsi pasif durumda olacak
End If
End Sub

```

'bu modül bizim formumuzun genişlediğinde elemanları form genişliğine ayarlar

```

Private Sub Form_Resize()
Static esk: Dim i, k, t
If IsEmpty(esk) Then esk = Form1.Width
If WindowState <> 1 Then
k = Form1.Width / esk
esk = Form1.Width
On Local Error Resume Next
For i = 0 To Form1.Count - 1
Controls(i).Left = Controls(i).Left * k

```

```
Controls(i).Width = Controls(i).Width * k
Next
End If
End Sub
```

```
'bu modül ise formu kapattığımızda portu kapatıyor
Private Sub Form_Unload(Cancel As Integer)
```

```
If MSComm1.PortOpen = True Then 'port açık değilse aç
```

```
MSComm1.PortOpen = False
```

```
End If
End Sub
```

```
'manuel moda geçiş.burada veri alışı durduruluyor ve elle kumanda ediliyor.tuşlar aktif
```

```
Private Sub Option1_Click()
If Option1.Value = True Then
Timer2.Enabled = False
cmdisit.Enabled = True
cmdsog.Enabled = True
Cmdstop.Enabled = True
End If
End Sub
```

```
'tekrar otomatik moda geçiş.
```

```
Private Sub Option2_Click()
If Option2.Value = True Then
MSComm1.InBufferCount = 0 'buffer sıfırla
Timer2.Enabled = True 'veri almaya başla
cmdisit.Enabled = False 'tuşlar pasif
cmdsog.Enabled = False
Cmdstop.Enabled = False
End If
End Sub
```

```
'off tuşunu aktif etmek için
```

```
Private Sub cmdoff_Click()
Timer4.Enabled = False 'on konumu veri gönderimi pasif
Timer5.Enabled = True 'off konumu veri gönderimi aktif
frmoff.BackColor = &HFF&
frmon.BackColor = &HFFFF&
End Sub
```

```
Private Sub Cmdon_Click()
Timer5.Enabled = False 'off konumu veri gönderimi pasif
Timer4.Enabled = True 'on konumu veri gönderimi aktif
frmon.BackColor = &HFF00&
frmoff.BackColor = &HFFFF&
End Sub
```

'manuel modda ısıtma butonu için yazılan modül

```
Private Sub cmdisit_Click()  
MSComm1.Output = "A"  
frmisit.BackColor = 192  
frmsog.BackColor = &HFFFFFF&  
End Sub
```

'manuel modda ısıtma-soğutma yi pasif etme için yazılan modül

```
Private Sub Cmdstop_Click()  
MSComm1.Output = "B"  
frmisit.BackColor = &HFFFFFF&  
frmsog.BackColor = &HFFFFFF&  
End Sub
```

'manuel modda soğutma butonu için yazılan modül

```
Private Sub cmdsoğ_Click()  
MSComm1.Output = "C"  
frmsog.BackColor = &HC00000  
frmisit.BackColor = &HFFFFFF&  
End Sub
```

'aşağıdaki yanıp sönen menü

```
Private Sub Timer1_Timer()  
Dim c  
c = Lblbaslik.BackColor  
Lblbaslik.BackColor = Lblbaslik.ForeColor  
Lblbaslik.ForeColor = c  
End Sub
```

'veri alışı ile ilgili modül

```
Private Sub Timer2_Timer()  
Dim i  
Dim m As Single  
Dim cevap  
i = Val(MSComm1.Input) 'i değerine veri okunuyor bu veri picden gelen değer  
m = (i * 0.4882) 'bu değeri 0.488 ile çarparak analog değere dönüştürdük  
'bu modül bizim look-up tablomuz bunlara göre karşılaştırma yapılıyor.  
If (0 < m) And (m <= 10.736) Then  
GoTo ilkon  
End If  
If (10.736 < m) And (m <= 20.496) Then  
GoTo ikion  
End If  
If (20.496 < m) And (m <= 30.286) Then  
GoTo üçon  
End If  
If (30.286 < m) And (m <= 40.016) Then  
GoTo dörton  
End If  
ilkon:  
If (1.6 <= m) And (m <= 2.4) Then  
Txtaksic.Text = 1  
GoTo son
```

```

End If
If (2.6 <= m) And (m <= 3.4) Then
Txtaksic.Text = 2
GoTo son
End If
If (3.6 <= m) And (m <= 4.4) Then
Txtaksic.Text = 3
GoTo son
End If

If (4.6 <= m) And (m <= 5.4) Then
Txtaksic.Text = 4
GoTo son
End If
If (5.6 <= m) And (m <= 6.4) Then
Txtaksic.Text = 5
GoTo son
End If
If (6.6 <= m) And (m <= 7.4) Then
Txtaksic.Text = 6
GoTo son
End If
If (7.6 <= m) And (m <= 8.4) Then
Txtaksic.Text = 7
GoTo son
End If
If (8.6 <= m) And (m <= 9.4) Then
Txtaksic.Text = 8
GoTo son
End If
If (9.6 <= m) And (m <= 10.4) Then
Txtaksic.Text = 9
GoTo son
End If
If (10.6 <= m) And (m <= 11.4) Then
Txtaksic.Text = 10
GoTo son
End If
ikion:
If (11.6 <= m) And (m <= 12.4) Then
Txtaksic.Text = 11
GoTo son
End If
If (12.6 <= m) And (m <= 13.4) Then
Txtaksic.Text = 12
GoTo son
End If
If (13.6 <= m) And (m <= 14.4) Then
Txtaksic.Text = 13
GoTo son

```

```

End If
If (14.6 <= m) And (m <= 15.4) Then
Txtaksic.Text = 14
GoTo son
End If
If (15.6 <= m) And (m <= 16.4) Then
Txtaksic.Text = 15
GoTo son
End If
If (16.6 <= m) And (m <= 17.4) Then
Txtaksic.Text = 16
GoTo son
End If
If (17.6 <= m) And (m <= 18.4) Then
Txtaksic.Text = 17
GoTo son
End If
If (18.6 <= m) And (m <= 19.4) Then
Txtaksic.Text = 18
GoTo son
End If
If (19.6 <= m) And (m <= 20.4) Then
Txtaksic.Text = 19
GoTo son
End If
If (20.6 <= m) And (m <= 21.4) Then
Txtaksic.Text = 20
GoTo son
End If
üçon:
If (21.6 <= m) And (m <= 22.4) Then
Txtaksic.Text = 21
GoTo son
End If
If (22.6 <= m) And (m <= 23.4) Then
Txtaksic.Text = 22
GoTo son
End If
If (23.6 <= m) And (m <= 24.4) Then
Txtaksic.Text = 23
GoTo son
End If
If (24.6 <= m) And (m <= 25.4) Then
Txtaksic.Text = 24
GoTo son
End If
If (25.6 <= m) And (m <= 26.4) Then
Txtaksic.Text = 25
GoTo son
End If

```

```

If (26.6 <= m) And (m <= 27.4) Then
Txtaksic.Text = 26
GoTo son
End If
If (27.6 <= m) And (m <= 28.4) Then
m = 27
Txtaksic.Text = m
GoTo son
End If
If (28.6 <= m) And (m <= 29.4) Then
m = 28
Txtaksic.Text = m
GoTo son
End If
If (29.6 <= m) And (m <= 30.4) Then
m = 29
Txtaksic.Text = m
GoTo son
End If
If (30.6 <= m) And (m <= 31.4) Then
Txtaksic.Text = 30
GoTo son
End If
dörton:
If (31.6 <= m) And (m <= 32.4) Then
Txtaksic.Text = 31
GoTo son
End If
If (32.6 <= m) And (m <= 33.4) Then
Txtaksic.Text = 32
GoTo son
End If
If (33.6 <= m) And (m <= 34.4) Then
Txtaksic.Text = 33
GoTo son
End If
If (34.6 <= m) And (m <= 35.4) Then
Txtaksic.Text = 34
GoTo son
End If
If (35.6 <= m) And (m <= 36.4) Then
Txtaksic.Text = 35
GoTo son
End If
If (36.6 <= m) And (m <= 37.4) Then
Txtaksic.Text = 36
GoTo son
End If
If (37.6 <= m) And (m <= 38.4) Then
Txtaksic.Text = 37

```

```
GoTo son
End If
If (38.6 <= m) And (m <= 39.4) Then
Txtaksic.Text = 38
GoTo son
End If
If (39.6 <= m) And (m <= 40.4) Then
Txtaksic.Text = 39
GoTo son
End If
If (40.6 <= m) And (m <= 41.4) Then
Txtaksic.Text = 40
GoTo son
End If
If (41.6 <= m) And (m <= 42.4) Then
Txtaksic.Text = 41
GoTo son
End If
If (42.6 <= m) And (m <= 43.4) Then
Txtaksic.Text = 42
GoTo son
End If
If (43.6 <= m) And (m <= 44.4) Then
Txtaksic.Text = 43
GoTo son
End If
If (44.6 <= m) And (m <= 45.4) Then
Txtaksic.Text = 44
GoTo son
End If
If (45.6 <= m) And (m <= 46.4) Then
Txtaksic.Text = 45
GoTo son
End If
If (46.6 <= m) And (m <= 47.4) Then
Txtaksic.Text = 46
GoTo son
End If
If (47.6 <= m) And (m <= 48.4) Then
Txtaksic.Text = 47
GoTo son
End If
If (48.6 <= m) And (m <= 49.4) Then
Txtaksic.Text = 48
GoTo son
End If
If (49.6 <= m) And (m <= 50.4) Then
Txtaksic.Text = 49
GoTo son
End If
```

```

If (50.6 <= m) And (m <= 51.4) Then
Txtaksic.Text = 50
GoTo son
End If
'look-up tablosu sonucu
son:
If Val(Txtaksic.Text) > Val(txtissic.Text) Then
Timer3.Enabled = True
End If
If Val(Txtaksic.Text) < Val(txtissic.Text) Then
Timer3.Enabled = True
End If
If Val(Txtaksic.Text) = Val(txtissic.Text) Then
Timer3.Enabled = True
End If
End Sub
'karşılaştırmalar yapıp PIC'e gönderme işleminin yapıldığı modül
Private Sub Timer3_Timer()
Dim renk
Dim renk1
'renk = frmsog.BackColor
'renk1 = frmisit.BackColor
'If txtissic.Text = "" Then
'frmsog.BackColor = renk
'frmisit.BackColor = renk1
'End If
If (Val(Txtaksic.Text) > Val(txtissic.Text)) And (txtissic.Text <> "") And (Timer4.Enabled =
True) Then
Timer5.Enabled = False
MSComm1.Output = "C"
frmsog.BackColor = &HC00000
frmisit.BackColor = &HFFFF&
Timer3.Enabled = False
End If
If (Val(Txtaksic.Text) > Val(txtissic.Text)) And (txtissic.Text <> "") And (Timer5.Enabled =
True) Then
Timer4.Enabled = False
txtissic.Text = ""
Txtaksic.Text = txtissic.Text
frmisit.BackColor = &HFFFF&
frmsog.BackColor = &HFFFF&
End If
If Val(Txtaksic.Text) < Val(txtissic.Text) And (Timer4.Enabled = True) Then
Timer5.Enabled = False
MSComm1.Output = "A"
frmisit.BackColor = 192
frmsog.BackColor = &HFFFF&
Timer3.Enabled = False
End If

```



```

If (Val(Txtaksic.Text) < Val(txtissic.Text)) And (txtissic.Text <> "") And (Timer5.Enabled =
True) Then
Timer4.Enabled = False
txtissic.Text = ""
Txtaksic.Text = txtissic.Text
frmisit.BackColor = &HFFFFFF&
frmisit.BackColor = &HFFFFFF&
frmsog.BackColor = &HFFFFFF&
End If
If Val(Txtaksic.Text) = Val(txtissic.Text) And (Timer4.Enabled = True) Then
Timer5.Enabled = False
MSComm1.Output = "B"
frmisit.BackColor = &HFFFFFF&
frmsog.BackColor = &HFFFFFF&
Timer3.Enabled = False
End If
If Val(Txtaksic.Text) = Val(txtissic.Text) And (Timer5.Enabled = True) Then
Timer4.Enabled = False
txtissic.Text = ""
Txtaksic.Text = txtissic.Text
frmisit.BackColor = &HFFFFFF&
frmisit.BackColor = &HFFFFFF&
frmsog.BackColor = &HFFFFFF&
End If
End Sub

Private Sub Timer4_Timer()
MSComm1.Output = "D"

End Sub

Private Sub Timer5_Timer()
MSComm1.Output = "E"
End Sub

```

6. KAYNAKÇA

Kitaplar

Mikrodenetleyiciler ve PIC programlama, Orhan Altınbaşak, Altaş Yayınları
PIC Basic Pro İle PIC Programlama, Orhan Altınbaşak, Altaş Yayınları
Her Yönüyle Seri Port, Jan Axelson
Elektronik Hobi, Güçlü Tuğay, Alfa Yayınları
Visual Basic 6.0 Pro, Türkmen kitabevi

Web Siteleri

<http://www.antrak.org.tr>
<http://www.bilimonline.8k.com>
<http://www.rccibitti.com>